AN ALTERNATIVE TO NONLINEAR ESTIMATION IN GUN FIRE CONTROL SYSTEMS

Robert Keith Brands

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

AN ALTERNATIVE TO NONLINEAR ESTIMATION
IN
GUN FIRE CONTROL SYSTEMS

by

Robert Keith Brands

December 1975

Thesis Advisor: D. E. Kirk

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) An Alternative to Nonlinear Estimation in Gun Fire Control Systems		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; December 1975
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) rands		8. CONTRACT OR GRANT NUMBER(s)
ON NAME AND ADDRESS iate School fornia 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
ME AND ADDRESS iate School fornia 93940		12. REPORT DATE December 1975
		13. NUMBER OF PAGES 149
E & ADDRESS(If different from Controlling Office) iate School fornia 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

(of this Report)

public release; distribution unlimited.

(of the abstract entered in Block 20, if different from Report)

IEEE Transactions on Aerospace and Electronic Systems, Vol 6, July 70.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Kalman Filter
Linear Estimator
Spherical Coordinate
Fire Control System

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A Kalman filter with assumed linear system dynamics was derived from a spherical coordinate state vector and applied to a fire control system tracking environment. Two models of target dynamics were developed: a constant-velocity model and a correlated random acceleration model. The predicted position performance of the filters derived from the two basic models was optimized using Monte Carlo methods

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601 |
(Page 1)

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

against a set of test trajectories that include constant-velocity and maneuvering target profiles. The results of the Monte Carlo simulation of these filters are compared with the results obtained from a filter derived from a Cartesian Coordinate state vector. Switch-on-range adaptive filters were developed from the two basic models and evaluated by Monte Carlo methods. The results of the simulations of the adaptive filters with the two different state vectors are compared.

An Alternative to Nonlinear Estimation in Gun Fire Control Systems

by

Robert Keith Brands
Lieutenant, United States Navy
B.S., Moorhead State College, 1969

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

ABSTRACT

A Kalman filter with assumed linear system dynamics was derived from a spherical coordinate state vector and applied to a fire control system tracking environment. Two models of target dynamics were developed: a constant-velocity model and a correlated random acceleration model. The predicted position performance of the filters derived from the two basic models was optimized using Monte Carlo methods against a set of test trajectories that include constant-velocity maneuvering target profiles. The results of the Monte Carlo simulation of these filters are compared with the results obtained from a filter derived from a Cartesian Coordinate state vector. Switch-on-range adaptive filters were developed from the two basic models and evaluated by Monte Carlo methods. The results of the simulations of the adaptive filters with the two different state vectors are compared.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

TABLE OF SYMBOLS AND ABBREVIATIONS

$\underset{\sim}{x}(K)$	n-dimensional state vector at time t = KT.
$\underset{\sim}{w}(K)$	p-dimensional random forcing vector at time t = KT.
$\underset{\sim}{\Phi}(K+1,K)$	(n x n)-dimensional state transition matrix.
$\underset{\sim}{\Gamma}(K)$	(n x p)-dimensional random forcing distribution matrix.
$\underset{\sim}{z}(K)$	m-dimensional vector of observations at time t = KT.
$\underset{\sim}{H}(K)$	(m x n)-dimensional observation matrix.
$\underset{\sim}{v}(K)$	m-dimensional measurement noise vector at time t = KT.
T	sampling period.
K	a non-negative integer denoting the current sample.
E	expectation operator.
δ_{kj}	Kronecker Delta Function.
$\underset{\sim}{Q}(K)$	(p x p)-dimensional covariance of random forcing matrix at time t = KT.
$\underset{\sim}{R}(K)$	(m x m)-dimensional covariance of measurement noise at time t = KT.
$\underset{\sim}{\hat{x}}(K/K)$	n-dimensional vector of state estimates at time t = KT, given measurements up to and including time t = KT.
$\underset{\sim}{\hat{x}}(K/K-1)$	n-dimensional vector of the one-step state prediction.
$\underset{\sim}{G}(K)$	(n x m)-dimensional Kalman Gain matrix at time t = KT.
$\underset{\sim}{P}(K/K)$	(n x n)-dimensional covariance of estimation error matrix.
$\underset{\sim}{P}(K+1/K)$	(n x n)-dimensional one-step prediction error covariance matrix.

$\underset{\sim}{I}$	(n x n)-dimensional identity matrix.
$[\ ]^T$	matrix(vector) transpose.
$[\ ]^{-1}$	matrix inverse.

I. INTRODUCTION

A. AN ALTERNATIVE TO NONLINEAR ESTIMATION IN GUN FIRE CONTROL SYSTEMS

The increased performance characteristics of modern aircraft and missiles has made apparent the need for more effective fire control systems. With the introduction of digital fire control computers, Kalman filtering techniques offer an attractive means of improving the effectiveness of a fire control system. Indeed, several recent studies conducted at the Naval Postgraduate School [Refs. 1-6] have specifically addressed the application of Kalman filtering techniques to fire control systems. It is interesting to note that the references cited above all have one distinctive feature in common; they use Cartesian Coordinate state vectors. A Cartesian Coordinate state vector offers the attraction of linear plant dynamics for a constant-velocity target but has the disadvantage of nonlinear measurement equations when used in a fire control system that measures the target's position in spherical coordinates. One possible solution to the nonlinear measurement problem is to process the sensor measurements through a prefilter as in Figure 1. The output of the sensor consists of noise corrupted measurements of the target's position in spherical coordinates

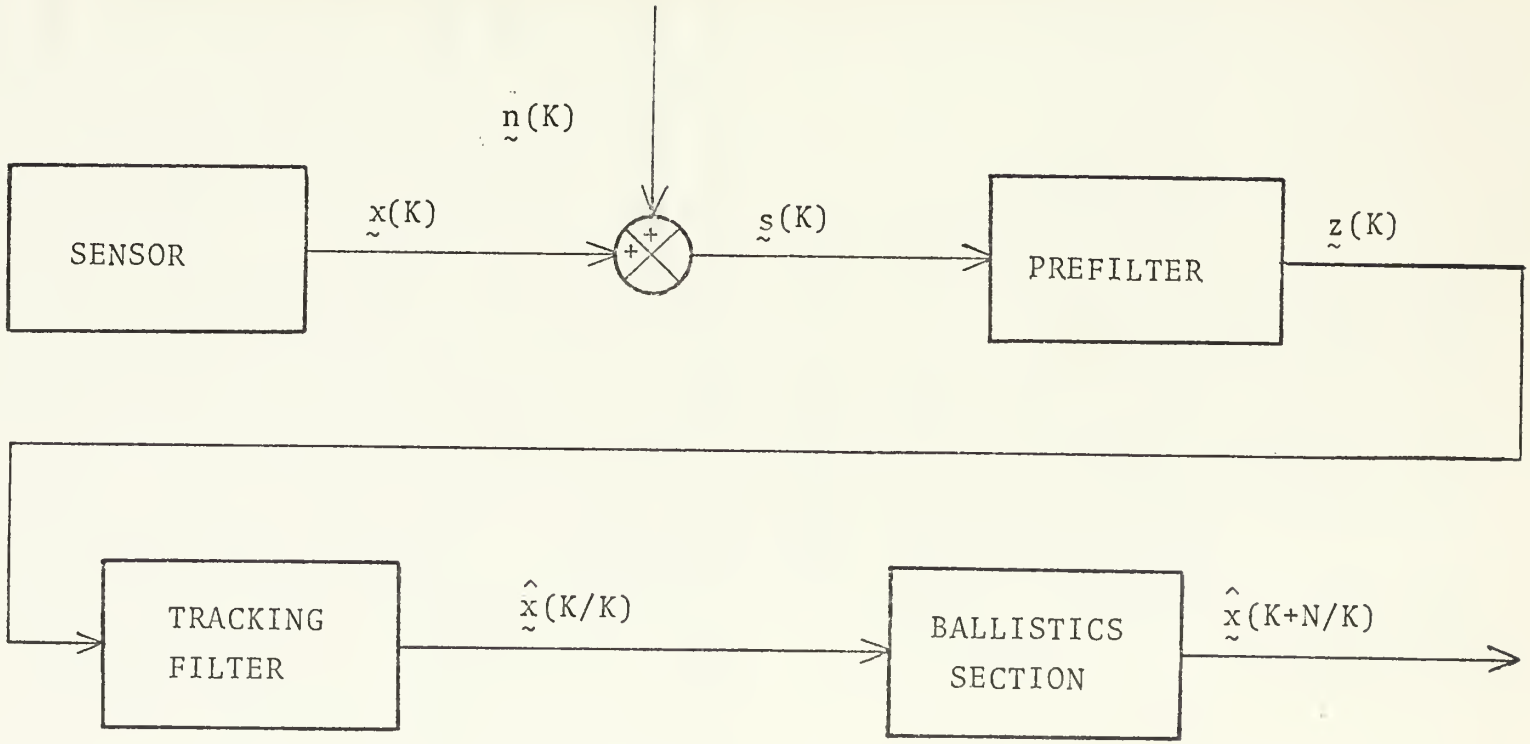


Fig. 1. FIRE CONTROL SYSTEM SIMPLIFIED BLOCK DIAGRAM

$$s_1(K) = r(K) + n_1(K)$$

$$s_2(K) = \phi(K) + n_2(K) \qquad (1.1)$$

$$s_3(K) = \theta(K) + n_3(K)$$

which serve as inputs to the prefilter. The appropriate coordinate transformation equations from Figure 2 are

$$x(K) = r(K)\sin\phi(K)\cos\theta(K)$$

$$y(K) = r(K)\cos\phi(K)\cos\theta(K) \qquad (1.2)$$

$$z(K) = r(K)\sin\theta(K)$$

where it has been shown [Ref. 3] that $\underset{\sim}{v}(K)$, the measurement noise process presented to the tracking filter is a non-linear, position-dependent function of the variances of the range, bearing, and elevation errors. Clearly then, the measurements become correlated in the coordinate conversion process. Full implementation of such a filtering scheme could conceivably involve much on-line computation since the statistics for $\underset{\sim}{v}(K)$ are no longer stationary. Another distinct disadvantage of the prefiltering method lies in the errors generated due to the added finite precision arithmetic of the coordinate conversion process. Such errors may be qualitatively described as a noise process whose statistical properties are difficult at best to identify. An excellent treatment of this specific topic is found in Gelb [Ref. 7].

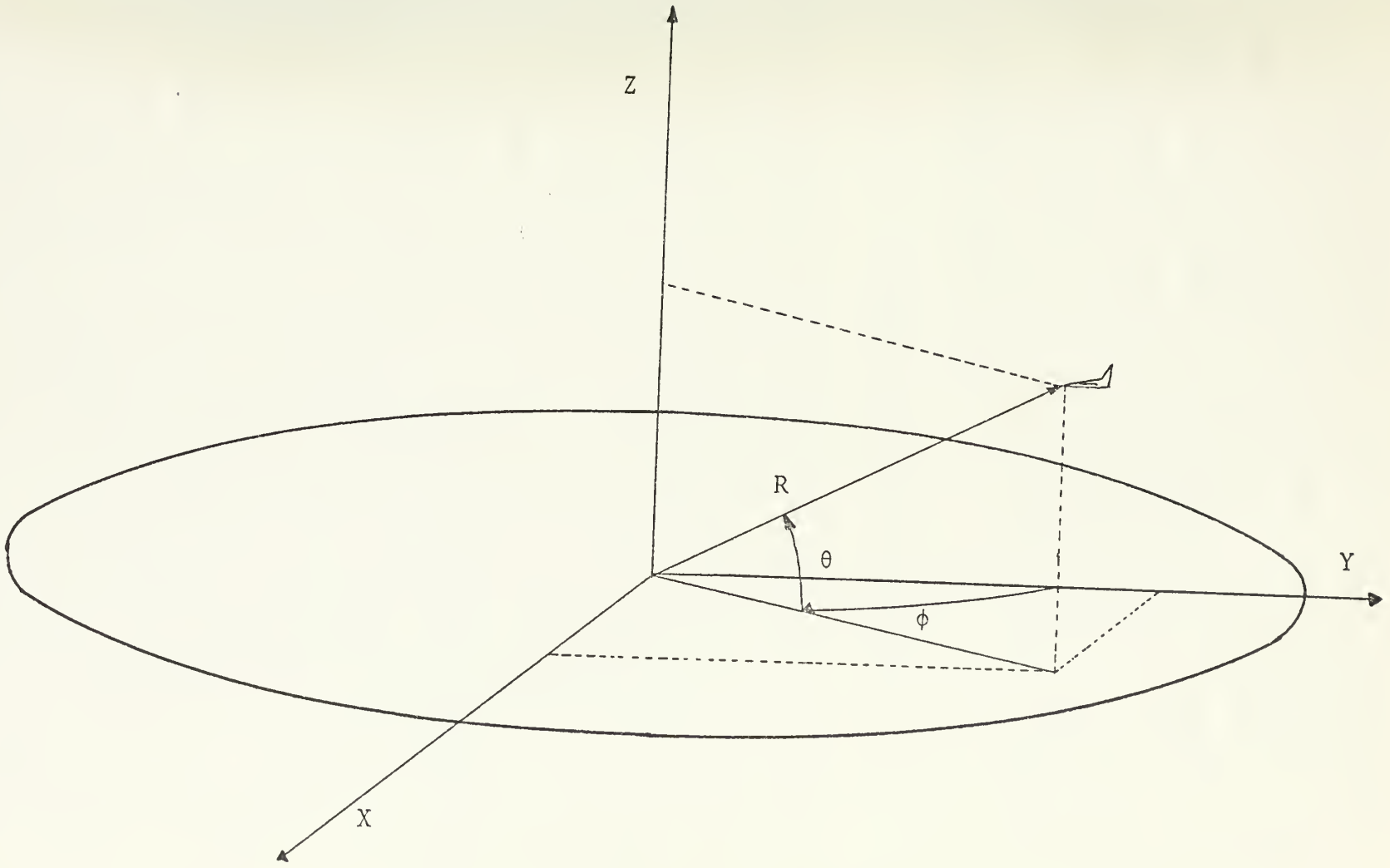


Fig. 2. THE SPHERICAL COORDINATE SYSTEM

Another approach to the nonlinear measurement process is to linearize the measurement equations about some reference or nominal trajectory. Unfortunately, such an approach is usually accomplished by the implementation of an extended or iterative extended Kalman filter with the attendant increase in on-line computing requirements.

Consider the filtering problem with a spherical coordinate state vector

$$x(K) = [r,\dot{r},\ddot{r},\phi,\dot{\phi},\ddot{\phi},\theta,\dot{\theta},\ddot{\theta}]^T \qquad (1.4)$$

Such a state vector has the advantage of linear measurement equations eliminating the correlation and position dependence of the measurement noise process. The disadvantage of a spherical state vector lies in the coupled and nonlinear system-target-dynamics, even for a constant-velocity target. Once again, the implementation of this filter would normally be with an extended or iterative extended Kalman filter. This thesis proposes a filter having a spherical coordinate state vector with assumed linear system dynamics. In view of the fact that any system model is an approximation to a physical system, this seems to be a reasonable assumption. Implementation of this would be a great deal simpler than the previously mentioned filters. A filter designed on this basis would then of necessity be sub-optimal. However, given acceptable performance, the sub-optimal filter might be preferred to the optimal nonlinear filter. Faced then,

with a sub-optimal filter due to modeling inaccuracies, a technique must be found to compensate for the unmodeled nonlinear properties of the system. Jazwinski [Ref. 8] has proposed two methods for the compensation of inaccurate a priori knowledge of the system. The first involves a fictitious noise input to the system, while the second overweights the most recent data. The net result of either method is an increase in the filter bandwidth, making the filter more responsive to system transients but increasing its sensitivity to the measurement noise process. The fictitious noise input method was chosen as the most promising and was used in this thesis. So that some quantifiable measure of the feasibility of these filters is possible, this thesis paralleled Parr's work [Ref. 4] and comparisons are made with his results.

B. LINEAR FILTERING THEORY

The system studied in this thesis is composed of a target and a sensor which gives noise-corrupted measurements of the target's position in spherical coordinates. The system may be approximated by the linear difference equations

$$\begin{aligned} \underset{\sim}{x}(K+1) &= \underset{\sim}{\phi}(K+1,K)\underset{\sim}{x}(K) + \underset{\sim}{\Gamma}(K)\underset{\sim}{w}(K) \\ \underset{\sim}{z}(K) &= \underset{\sim}{H}(K)\underset{\sim}{x}(K) + \underset{\sim}{v}(K) \end{aligned} \tag{1.5}$$

The statistics assumed for the stochastic processes $\underset{\sim}{w}(K)$

and $\underset{\sim}{v}(K)$ are

$$E[\underset{\sim}{w}(K)] = E[\underset{\sim}{v}(K)] = \underset{\sim}{0} \quad \text{for all } K \tag{1.6}$$

$$E[\underset{\sim}{v}(k)\underset{\sim}{v}^T(j)] = \underset{\sim}{R}(K)\delta_{kj} \tag{1.7}$$

$$E[\underset{\sim}{w}(k)\underset{\sim}{w}^T(j)] = \underset{\sim}{Q}(K)\delta_{kj} \tag{1.8}$$

$$E[\underset{\sim}{w}(K)\underset{\sim}{v}^T(K)] = E[\underset{\sim}{v}(K)\underset{\sim}{w}^T(K)] = \underset{\sim}{0} \quad \text{for all } K \tag{1.9}$$

$$E[\underset{\sim}{w}(K)\underset{\sim}{x}^T(0)] = E[\underset{\sim}{x}(0)\underset{\sim}{w}^T(K)] = \underset{\sim}{0} \quad \text{for all } K \tag{1.10}$$

$$E[\underset{\sim}{v}(K)\underset{\sim}{x}^T(0)] = E[\underset{\sim}{x}(0)\underset{\sim}{v}^T(K)] = \underset{\sim}{0} \quad \text{for all } K \tag{1.11}$$

The linear, discrete Kalman filter is characterized by the difference equations

$$\hat{\underset{\sim}{x}}(K/K) = \hat{\underset{\sim}{x}}(K/K-1) + \underset{\sim}{G}(K)[\underset{\sim}{z}(K) - \underset{\sim}{H}(K)\hat{\underset{\sim}{x}}(K/K-1)] \tag{1.12}$$

with the one-step prediction given by

$$\hat{\underset{\sim}{x}}(K/K-1) = \underset{\sim}{\phi}(K,K-1)\hat{\underset{\sim}{x}}(K-1/K-1) \tag{1.13}$$

The Kalman Gain matrix, $\underset{\sim}{G}(K)$, is found by solving the following equations

$$\underset{\sim}{G}(K) = \underset{\sim}{P}(K/K-1)\underset{\sim}{H}^T(K)[\underset{\sim}{H}(K)\underset{\sim}{P}(K/K-1)\underset{\sim}{H}^T(K) + \underset{\sim}{R}(K)]^{-1} \tag{1.14}$$

$$\underset{\sim}{P}(K/K) = [\underset{\sim}{I} - \underset{\sim}{G}(K)\underset{\sim}{H}(K)]\underset{\sim}{P}(K/K-1) \tag{1.15}$$

$$\underset{\sim}{P}(K+1/K) = \underset{\sim}{\phi}(K+1,K)\underset{\sim}{P}(K/K)\underset{\sim}{\phi}^T(K+1,K) + \underset{\sim}{\Gamma}(K)\underset{\sim}{Q}(K)\underset{\sim}{\Gamma}^T(K) \tag{1.16}$$

To start the gain calculations, the matrix $\underset{\sim}{P}(K/K-1)$ must

be initialized. Taking the standard approach, $\underset{\sim}{P}(0/-1)$ is given by

$$\underset{\sim}{P}(0/-1) = E[(\underset{\sim}{x}(0) - \bar{\underset{\sim}{x}}_0)(x(0) - \bar{\underset{\sim}{x}}_0)^T] \tag{1.17}$$

where

$$E[\underset{\sim}{x}(0)] = \bar{\underset{\sim}{x}}_0 \tag{1.18}$$

An unbiased estimator is obtained if the filter is started by setting

$$\hat{\underset{\sim}{x}}(0/-1) = \bar{\underset{\sim}{x}}_0 \tag{1.19}$$

II. DEVELOPMENT OF THE FILTER MODELS

A. CONSTANT-VELOCITY ($1/S^2$) MODEL

The first filter tested was based on a dynamic system model that assumes linear constant-velocity target motion in each of three coordinate directions.

The state transition matrix $\underset{\sim}{\phi}(K+1,K)$ and the random forcing input distribution matrix $\underset{\sim}{\Gamma}(K)$ are

$$\underset{\sim}{\phi}(K+1,K) = \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

and

$$\underset{\sim}{\Gamma}(K) = \begin{bmatrix} \frac{T^2}{2} & 0 & 0 \\ T & 0 & 0 \\ 0 & \frac{T^2}{2} & 0 \\ 0 & T & 0 \\ 0 & 0 & \frac{T^2}{2} \\ 0 & 0 & T \end{bmatrix} \tag{2.2}$$

where T is the sampling period.

Since the state vector is expressed in spherical coordinates, the measurement process is linear and is given by

$$\underset{\sim}{z}(K) = \underset{\sim}{H}(K)\underset{\sim}{x}(K) + \underset{\sim}{v}(K) \tag{2.3}$$

The measurement noise processes for the individual coordinate directions are assumed to be independent processes. The covariance of measurement noise matrix $\underset{\sim}{R}(K)$ is diagonal and is given by

$$\underset{\sim}{R}(K) = \begin{bmatrix} \sigma_R^2 & & \underset{\sim}{0} \\ & \sigma_B^2 & \\ \underset{\sim}{0} & & \sigma_E^2 \end{bmatrix} \tag{2.4}$$

Because of its simplicity the constant-velocity model cannot realistically be expected to perform well against all types of target trajectories and may not perform acceptably against maneuvering targets. The performance of the constant-velocity model filter can be improved by the use of the system random input covariance matrix $\underset{\sim}{Q}(K)$. Indeed, one of the functions of the $\underset{\sim}{Q}(K)$ matrix is to compensate for the inadequacies of a simplified model, and for this specific thesis, $\underset{\sim}{Q}(K)$ can also partially account for the system nonlinearities not explicitly included in the system model. The subject of the sensitivity of the

filter's N-step prediction performance to the $\underset{\sim}{Q}(K)$ matrix forms the basis for the simulations of Chapter III.

Initialization of the filter requires a priori knowledge of the statistics of the system initial conditions so that the gain calculations may be started. For any fixed coordinate system, it is reasonable to consider the mean of all possible system initial conditions as zero or

$$E[\underset{\sim}{x}(0)] = \underset{\sim}{\bar{x}}_0 = \underset{\sim}{0} \tag{2.5}$$

It has been shown [Refs. 7 and 9] that under proper initialization, the filter will remain unbiased for $K \geq 0$, i.e.,

$$E[\underset{\sim}{e}(K/K)] = \underset{\sim}{0} \quad \text{for all } K \tag{2.6}$$

where

$$\underset{\sim}{e}(K/K) = \underset{\sim}{\hat{x}}(K/K) - \underset{\sim}{x}(K) \tag{2.7}$$

The unbiased condition is satisfied if

$$\underset{\sim}{\hat{x}}(0/-1) = \underset{\sim}{\bar{x}}_0 \tag{2.8}$$

Initialization of the covariance of estimation error matrix requires a somewhat different approach. Following Parr [Ref. 4] it is quantitatively shown that if the diagonal elements of the initial covariance of estimation error matrix are made very large, the net effect is that the first position estimate is made identically equal to

the first system measurement. In this manner the system is forced into tracking the target rapidly. For all simulations conducted then, it was assumed that

$$\underset{\sim}{P}(0/-1) = \begin{bmatrix} 10^8 & & \underset{\sim}{0} \\ & \ddots & \\ \underset{\sim}{0} & & 10^8 \end{bmatrix} \tag{2.9}$$

B. CORRELATED RANDOM ACCELERATION MODEL

The constant-velocity model discussed in the previous section views any target maneuvers as small perturbations about a nominal, constant-velocity trajectory. This tacitly assumes that any target maneuvers are uncorrelated in time, or from one measurement to the next. Target maneuvers will generally take place over a period of n measurements and the magnitude of n is dependent on the level of the target's maneuver, i.e., for a slow turn n is relatively large whereas for an evasive type maneuver n is smaller. The fact that the accelerations occur over a finite period of time correlates the target's accelerations (maneuvers) in time. One approach to modeling the effects of time-correlated target maneuvers is to use a constant-velocity plant driven by correlated noise obtained from a low-pass filter of the form

$$T(s) = \frac{\alpha}{(s + \alpha)} \tag{2.10}$$

driven by white noise. The constant α is interpreted as the time constant or degree of time correlation of the target's maneuver.

For one dimension, the overall model transfer function is

$$T(s) = \frac{\alpha}{s^2(s + \alpha)} \tag{2.11}$$

as depicted in Figure 3.

For the three-dimensional case, the filter is ninth order and has the state vector

$$\underset{\sim}{x}(K) = [r, \dot{r}, \ddot{r}, \phi, \dot{\phi}, \ddot{\phi}, \theta, \dot{\theta}, \ddot{\theta}]^T \tag{2.12}$$

The correlated random acceleration model is discussed in detail by McKinley [Ref. 5] and Singer [Ref. 10]. From the cited references, the (3 x 3) state transition matrix $\underset{\sim}{\phi}_i(T)$ and the (3 x 1) random forcing input vector $\underset{\sim}{\Gamma}_i(T)$ are for the single dimension with assumed linear system dynamics

$$\phi_i(T) = \begin{bmatrix} 1.0 & T & \frac{T}{\alpha} - \frac{1}{\alpha^2}(1 - e^{-\alpha T}) \\ 0.0 & 1.0 & \frac{1}{\alpha}(1 - e^{-\alpha T}) \\ 0.0 & 0.0 & e^{-\alpha T} \end{bmatrix} \tag{2.13}$$

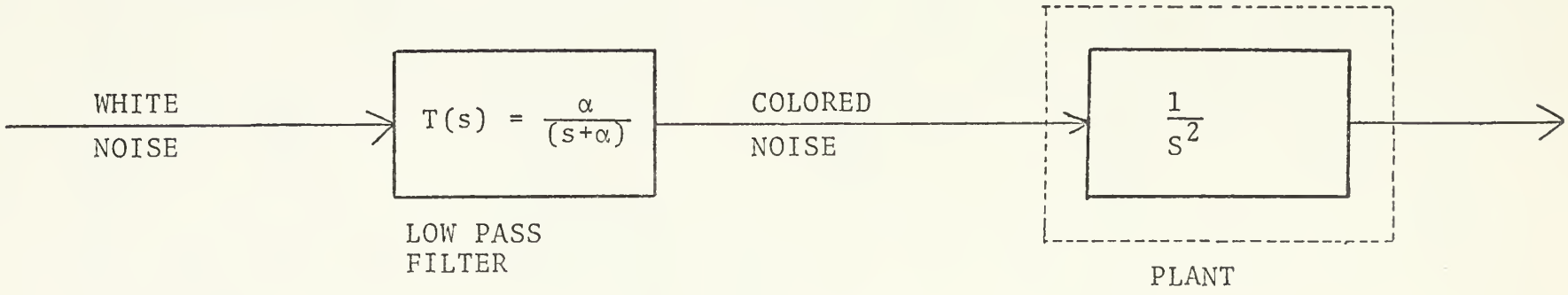


Fig. 3. CONSTANT-VELOCITY PLANT DRIVEN BY CORRELATED NOISE

and

$$\underset{\sim}{\Gamma}_i(T) = \begin{bmatrix} \frac{T^2}{2} - \frac{T}{\alpha} + \frac{1}{\alpha^2}(1 - e^{-\alpha T}) \\ T - \frac{1}{\alpha}(1 - e^{-\alpha T}) \\ (1 - e^{-\alpha T}) \end{bmatrix} \tag{2.14}$$

For the ninth-order filter the state transition matrix $\underset{\sim}{\phi}(K+1,K)$ is formed by

$$\underset{\sim}{\phi}(K+1,K) = \begin{bmatrix} \underset{\sim}{\phi}_r(T) & \underset{\sim}{0} & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{\phi}_B(T) & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{0} & \underset{\sim}{\phi}_E(T) \end{bmatrix} \tag{2.15}$$

where $\underset{\sim}{\phi}_r(T)$, $\underset{\sim}{\phi}_B(T)$, and $\underset{\sim}{\phi}_E(T)$ are individually given by $\underset{\sim}{\phi}_i(T)$ in Equation (2.13). In a similar manner, for the ninth-order filter the three-dimensional system matrix $\underset{\sim}{\Gamma}(K)$ is given by

$$\underset{\sim}{\Gamma}(K) = \begin{bmatrix} \underset{\sim}{\Gamma}_r(T) & \underset{\sim}{0} & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{\Gamma}_B(T) & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{0} & \underset{\sim}{\Gamma}_E(T) \end{bmatrix} \tag{2.16}$$

where $\underset{\sim}{\Gamma}_r(T)$, $\underset{\sim}{\Gamma}_B(T)$, $\underset{\sim}{\Gamma}_E(T)$ are individually given by $\underset{\sim}{\Gamma}_i(T)$ in Equation (2.14).

The covariance of measurement noise matrix $\underset{\sim}{R}(K)$ is the same as given in the preceding section by Equation (2.4). Initialization of the filter is carried out in exactly the same fashion as described in the previous section by Equations (2.5), (2.8), and (2.9). As in the preceding section, $\underset{\sim}{Q}(K)$, the covariance of the random forcing input white noise process presented to the coloring filter, was used to compensate for the unmodeled dynamic nonlinearities. The sensitivity of the filter's N-step prediction performance to the $\underset{\sim}{Q}(K)$ matrix selected is treated in Chapter III.

In the interest of saving computer time, the correlated random acceleration model was tested using a single value for α, the maneuver time constant. Since Parr [Ref. 4] reported that the best overall performance in his simulations was obtained from a value of $\alpha = 0.5$, that value was used in the study of this model. Additionally, the maneuver time constant α, was fixed at the same value for each of the three coordinate directions, though this constraint need not necessarily be imposed on the system.

III. RESULTS OF THE MONTE CARLO SIMULATION OF THE PROPOSED LINEAR FILTERS

A. EXPERIMENTAL PROCEDURE

Monte Carlo simulation techniques were used in the optimization and testing of the filters developed in Chapter II. A listing and description of the FORTRAN program used is given in Appendix A. All results are the products of 100-member ensemble simulations. Test trajectories 1-8 were used in the optimization and testing process and Appendix B gives a description of the profile for each trajectory. The profiles of trajectories 1, 2, 3, and 4 are identical to the short-range constant-velocity, short-range maneuvering, long-range constant-velocity, and long-range maneuvering profiles, respectively, used by Parr [Ref. 4]. Trajectories 5-8 were included to examine the filter's performance against more severe target maneuvers.

The elements of $\underset{\sim}{R}(K)$, the covariance of measurement noise process matrix, were fixed for all simulations at

$$\sigma_R = 3.0 \text{ yds.}$$
$$\sigma_B = 0.01722 \text{ Deg.} \qquad (3.1)$$
$$\sigma_E = 0.01722 \text{ Deg.}$$

The values given in Equation (3.1) are those used by Parr [Ref. 4] and are used in this thesis in an attempt to

duplicate Parr's testing environment, and minimize procedural discrepancies.

The performance evaluation of the filters developed in Chapter II is based on their N-step prediction or position-state prediction accuracy. A detailed description of the method used in the determination of the error in the predicted position is given in Appendix C. The results presented in this chapter are the percentages of "hits" and "scares" computed over the entire ensemble. A "hit" is defined as an error in the predicted position of less than eight yards, while a "scare" is defined as an error in the predicted position of less than 16 yards, but greater than eight yards.

Initialization of each filter simulated was identical in all trials and is described in Chapter II by Equations (2.5), (2.8), and (2.9).

The elements of the covariance of the random forcing input matrix $\underset{\sim}{Q}(K)$ were determined by a two-dimensional optimization search. The assumption of linear uncoupled dynamics eliminates any spatial correlation, resulting in a diagonal $\underset{\sim}{Q}(K)$ matrix of the form

$$\underset{\sim}{Q}(K) = \begin{bmatrix} q_r & 0 & 0 \\ 0 & q_B & 0 \\ 0 & 0 & q_E \end{bmatrix} \tag{3.2}$$

If the system dynamics were truly linear, then the diagonal elements of (3.2) would represent the variances of the random forcing input to the system. However, since $\underset{\sim}{Q}(K)$ is also used to compensate for inaccuracies in the system model, q_r, q_B and q_E must be adjusted accordingly. To simplify the optimization process for both models, q_B and q_E were constrained to be equal throughout. The two-dimensional search optimized the filter's position-state prediction accuracy as a function of two variables, q_r and the pair $q_B = q_E$.

In the case of the constant-velocity model, the elements of the matrix $\underset{\sim}{Q}(K)$ serve as a means by which the bandwidth of the filter is increased to ensure that the filter will respond to target accelerations. This, of course, is in addition to compensation for the unmodeled nonlinearities of the system dynamics. Because of the dual role played by the matrix $\underset{\sim}{Q}(K)$, a brute force search was done to determine the elements of $\underset{\sim}{Q}(K)$ that gave the best position-state prediction performance. In a similar manner, for the filter based on the correlated random acceleration model, the bandwidth must be increased to permit sufficient response to targets exhibiting non-zero acceleration rates. Here again brute force search techniques were used to determine the $\underset{\sim}{Q}(K)$ matrix that gave the best position prediction performance.

For both models, once the optimum $\underset{\sim}{Q}(K)$ matrix was identified for a specific trajectory, that particular

filter configuration was tested against the remaining trajectories. The net result being that each optimized filter was tested against seven additional trajectories to ensure that the optimum $\underset{\sim}{Q}(K)$ had been found, and to determine the effect of using a $\underset{\sim}{Q}(K)$ selected for a track of one type against targets having different maneuvering characteristics.

B. SIMULATION RESULTS FOR THE CONSTANT-VELOCITY (L/S^2) MODEL

The results presented in this section are for the optimized filter configurations. The term "Filter 1" denotes the filter with the specific $\underset{\sim}{Q}(K)$ matrix that gave the best position prediction performance against trajectory 1. Thus, the simulation results for eight filters, or more precisely, eight filter configurations are presented.

As pointed out in the preceding section, the assumption of linear uncoupled dynamics results in the diagonal $\underset{\sim}{Q}(K)$ matrix given by Equation (3.2). Table I gives the diagonal elements of the $\underset{\sim}{Q}(K)$ matrix that provided the best position prediction performance for each test trajectory. The relative magnitude of the diagonal elements for the optimal filters are as expected. That is to say, the elements of $\underset{\sim}{Q}(K)$ for the non-maneuvering trajectories (1, 3, 7, 8) are lower in magnitude than those for the maneuvering trajectories (2, 4, 5, 6). Similarly, the elements of $\underset{\sim}{Q}(K)$ for the long-range maneuvering trajectories are lower in magnitude than those for the short-range maneuvering trajectories. Similar

Table I

DIAGONAL ELEMENTS OF THE OPTIMAL $\underset{\sim}{Q}(K)$ MATRICES FOR THE CONSTANT-VELOCITY ($1/S^2$) MODEL

FILTER	q_{11}	q_{22}	q_{33}
1	1.00×10^{-6}	8.58×10^{-9}	8.58×10^{-9}
2	0.05	0.1153	0.1153
3	1.00×10^{-6}	1.13×10^{-10}	1.13×10^{-10}
4	0.025	0.006553	0.006553
5	0.0075	0.1372	0.1372
6	1.00×10^{-4}	5.55×10^{-7}	5.55×10^{-7}
7	0.01	3.37×10^{-7}	3.37×10^{-7}
8	0.005	1.66×10^{-7}	1.66×10^{-7}

observations may be made for the moderate level maneuvers as opposed to the evasive level maneuvers. Note that the elements of $\underset{\sim}{Q}(K)$ for the long-range evasive maneuver profile (trajectory 6) are smaller than those for the long-range moderate level maneuver profile (trajectory 4). This appears to be a reasonable phenomenon in light of the fact that overweighting the estimates for a faster more highly maneuvering target at long range could result in larger errors in the predicted position. It follows then that the bandwidth of the optimal filter for the long-range evasive maneuver should be somewhat smaller than the bandwidth for the moderate level maneuver at long range.

Table II gives the simulation results for the optimal filter configurations as well as for each optimal filter configuration against the other trajectories. The results shown are not surprising in that for a specific trajectory, the optimal filter configuration gave the best results. In a few isolated cases, other filter configurations gave results that approach those for the optimal configuration. Though this was not expected, the differences are not considered small enough to justify any additional effort in the optimization process.

Table III compares the hit/scare percentages from the simulations of the two models using the two different state vectors. The results for the spherical coordinate state vector represent the optimal performance against a specific

Table II

SIMULATION RESULTS FOR THE OPTIMIZED
CONSTANT-VELOCITY MODEL FILTERS

FILTER	Trajectory 1		Trajectory 2		Trajectory 3		Trajectory 4	
	%hit	%scare	%hit	%scare	%hit	%scare	%hit	%scare
1	86.89	5.00	9.02	11.45	40.21	22.29	0.0	0.10
2	71.77	16.48	25.34	24.05	0.71	3.62	0.1	0.86
3	86.86	4.91	9.02	11.45	40.81	21.14	0.0	0.10
4	85.82	5.84	20.57	26.11	5.10	19.10	0.71	0.95
5	70.34	17.27	24.84	24.66	0.0	4.76	0.0	0.0
6	85.23	4.68	9.02	11.45	40.03	21.32	0.0	0.0
7	86.59	4.91	9.02	11.45	40.24	22.43	0.0	0.0
8	86.62	4.93	9.02	11.45	40.38	22.29	0.0	0.0

Table II

SIMULATION RESULTS FOR THE OPTIMIZED

CONSTANT-VELOCITY MODEL FILTERS (Cont'd)

FILTER	Trajectory 5		Trajectory 6		Trajectory 7		Trajectory 8	
	%hit	%scare	%hit	%scare	%hit	%scare	%hit	%scare
1	12.17	16.05	7.00	0.14	85.22	5.35	69.10	11.36
2	27.62	11.64	0.46	1.57	61.80	20.17	3.57	14.29
3	12.43	15.79	7.00	0.14	86.07	5.35	69.09	11.36
4	28.95	12.40	0.04	0.11	82.70	8.62	27.29	37.25
5	29.33	12.17	0.0	0.0	59.87	20.72	4.25	12.71
6	12.17	16.02	8.13	0.11	85.14	5.36	68.42	11.27
7	11.95	16.24	6.96	0.0	86.22	5.20	69.05	11.39
8	11.69	16.52	7.00	0.0	86.14	5.17	69.20	12.21

Table III

COMPARISON OF RESULTS FOR THE SPHERICAL AND CARTESIAN COORDINATE STATE VECTOR CONSTANT-VELOCITY MODELS

STATE VECTOR	CONSTANT VELOCITY TRAJECTORIES				MANEUVERING TRAJECTORIES			
	SHORT-RANGE		LONG-RANGE		SHORT-RANGE		LONG-RANGE	
	%hit	%scare	%hit	%scare	%hit	%scare	%hit	%scare
SPHERICAL COORDINATE	86.89	5.00	40.81	21.14	25.34	24.05	0.71	0.95
CARTESIAN COORDINATE	88.4	3.00	34.7	24.7	15.20	13.30	0.33	1.10

trajectory. The results tabulated for the Cartesian coordinate state vector are the best of the results reported by Parr [Ref. 4].

Tables IV and V show the tracking performance of each optimal filter against its specific trajectory. For example, the tracking performance of filter 1 against trajectory 1 is given, and so on. The tabulated data represent the means of estimation error, defined by Equation (2.7), time averaged over the entire trajectory, and the RMS value of the deviation of the means of estimation error about their time average.

C. SIMULATION RESULTS FOR THE CORRELATED RANDOM ACCELERATION MODEL

The results presented in this section are for the optimized filter configurations. Since the optimization search for the ninth-order filter required a great deal more computer time, this model was only optimized against trajectories 1-3. As in the preceding section, the term "Filter 1A" denotes the filter with the specific $\underset{\sim}{Q}(K)$ matrix that gave the best position prediction performance against trajectory 1.

The tabulated data of Table VI gives the elements of the $\underset{\sim}{Q}(K)$ matrices that provided the optimum performance against each test trajectory. The data appear to be consistent within the tracking environments presented to the filter. Notice that in each configuration the elements of the optimum $\underset{\sim}{Q}(K)$ matrices for the correlated random

Table IV

POSITION ESTIMATION PERFORMANCE OF THE

OPTIMIZED CONSTANT-VELOCITY MODEL FILTERS

FILTER	$\bar{e}_r$(yds)	$\sigma_{\bar{e}_r}$(yds)	$\bar{e}_\phi$(deg)	$\sigma_{\bar{e}_\phi}$(deg)	$\bar{e}_\theta$(deg)	$\sigma_{\bar{e}_\theta}$(deg)
1	5.33×10^{-2}	0.126	1.52×10^{-4}	6.18×10^{-4}	-2.67×10^{-4}	8.11×10^{-4}
2	-8.70×10^{-2}	0.282	2.32×10^{-3}	1.06×10^{-2}	2.90×10^{-6}	1.50×10^{-3}
3	4.95×10^{-2}	0.126	1.52×10^{-4}	6.18×10^{-4}	-2.68×10^{-4}	8.11×10^{-4}
4	-7.00×10^{-2}	0.219	1.69×10^{-3}	1.32×10^{-2}	5.22×10^{-6}	1.15×10^{-3}
5	9.04×10^{-2}	0.985	2.35×10^{-4}	1.95×10^{-2}	1.47×10^{-5}	1.56×10^{-3}
6	2.93×10^{-2}	0.894	-2.85×10^{-2}	2.93×10^{-1}	-3.77×10^{-4}	9.46×10^{-4}
7	6.00×10^{-2}	0.127	1.40×10^{-4}	6.21×10^{-4}	-2.50×10^{-4}	8.10×10^{-4}
8	5.25×10^{-2}	0.125	1.39×10^{-4}	6.19×10^{-4}	-2.60×10^{-4}	8.10×10^{-4}

Table V

RATE ESTIMATION PERFORMANCE OF THE

OPTIMIZED CONSTANT-VELOCITY MODEL FILTERS

FILTER	$\bar{e}_{\dot{r}}$ (y/s)	$\sigma_{\bar{e}_{\dot{r}}}$ (y/s)	$\bar{e}_{\dot{\phi}}$ (d/s)	$\sigma_{\bar{e}_{\dot{\phi}}}$ (d/s)	$\bar{e}_{\dot{\theta}}$ (d/s)	$\sigma_{\bar{e}_{\dot{\theta}}}$ (d/s)
1	3.79	29.70	1.00×10^{-5}	5.26×10^{-4}	-2.45×10^{-4}	1.57×10^{-3}
2	3.69	29.72	3.36×10^{-2}	1.90×10^{-1}	-5.45×10^{-4}	5.71×10^{-3}
3	3.80	29.70	9.97×10^{-6}	5.26×10^{-4}	-2.45×10^{-4}	1.57×10^{-3}
4	3.70	29.71	3.75×10^{-3}	6.67×10^{-2}	-1.48×10^{-4}	2.18×10^{-3}
5	4.99	38.75	-2.51×10^{-2}	2.36×10^{-1}	-1.62×10^{-4}	5.77×10^{-3}
6	4.98	38.75	-1.15×10^{-2}	2.86×10^{-1}	-2.97×10^{-4}	1.68×10^{-3}
7	4.96	38.74	6.76×10^{-5}	5.27×10^{-4}	-2.38×10^{-4}	1.57×10^{-3}
8	4.97	38.73	6.67×10^{-5}	5.27×10^{-4}	-2.42×10^{-4}	1.57×10^{-3}

Table VI

DIAGONAL ELEMENTS OF THE OPTIMIZED $\underset{\sim}{Q}(K)$ MATRICES FOR THE CORRELATED RANDOM ACCELERATION MODEL FILTERS

FILTER	q_{11}	q_{22}	q_{33}
1A	1.00×10^{-7}	5.00×10^{-9}	5.00×10^{-9}
2A	0.100	14.599	14.599
3A	1.00×10^{-3}	1.46×10^{-7}	1.46×10^{-7}

acceleration model are larger in magnitude than those for the corresponding entries of the constant-velocity model. This seems reasonable because this filter is required to respond to a transient (due to the non-zero acceleration rate) that is expected to be smaller in magnitude than the transient encountered by the constant-velocity model filter. For this reason the bandwidth of this filter must be increased to ensure sufficient response to the non-zero acceleration rates. During the optimization search, it was observed that, with one exception, this filter model is extremely sensitive to variations of elements q_{22} and q_{33} corresponding to acceleration rates normal to the Line-of-Sight. This is to be expected, since such accelerations cause the greatest displacement from the nominal trajectory adding a source for potentially large errors in the predicted position. The single exception noted was for trajectory 1. In this specific case the filter was surprisingly insensitive to variations in the elements of the $\underset{\sim}{Q}(K)$ matrix, and its predicted position response was fairly flat over a four-decade range (10^{-11} - 10^{-7}). The elements reported for Filter 1A are therefore somewhat arbitrary, lying in the center of the flat region. Values for the pair $q_{22} = q_{33}$ above and below this region produced inferior results. A refined search grid in the flat region failed to produce any significant change in the results reported for Filter 1A.

Table VII gives the simulation results for each optimal filter configuration and for each optimal filter configuration against the two remaining trajectories. Table VIII compares the hit/scare percentages from the simulations of the models using the two different state vectors. The data presented for the spherical coordinate state vector represent the optimum performance for the trajectories considered. The results reported for the Cartesian Coordinate state vector model represent the best of the results reported by Parr [Ref. 4]. Tables IX through XI show the tracking performance for each optimal filter configuration against its specific trajectory. The results shown are the time-averaged means of estimation error and the RMS value of the deviation of the means of estimation error from their time average.

Table VII

SIMULATION RESULTS FOR THE OPTIMIZED
CORRELATED RANDOM ACCELERATION MODELS

FILTER	TRAJECTORY 1		TRAJECTORY 2		TRAJECTORY 3	
	%hit	%scare	%hit	%scare	%hit	%scare
1A	74.41	7.27	7.20	8.27	8.95	19.48
2A	32.89	15.57	26.43	14.89	0.0	0.05
3A	74.09	7.55	7.16	8.27	12.93	18.52

Table VIII

COMPARISON OF RESULTS FOR THE SPHERICAL AND CARTESIAN

COORDINATE STATE VECTOR CORRELATED RANDOM ACCELERATION MODELS

STATE VECTOR	CONSTANT-VELOCITY TRAJECTORIES				MANEUVERING TRAJECTORIES			
	SHORT-RANGE		LONG-RANGE		SHORT-RANGE		LONG-RANGE	
	%hit	%scare	%hit	%scare	%hit	%scare	%hit	%scare
SPHERICAL COORDINATE	74.41	7.27	12.93	18.52	26.43	14.89	NOT TESTED	
CARTESIAN COORDINATE	73.40	7.70	6.60	17.40	21.20	9.00	NOT TESTED	

Table IX

POSITION ESTIMATION PERFORMANCE OF THE OPTIMIZED CORRELATED RANDOM ACCELERATION MODEL FILTERS

PARAMETER	FILTER 1A	2A	3A
$\bar{e}_r$ (yds)	5.87×10^{-2}	0.111	1.06×10^{-2}
$\sigma_{\bar{e}_r}$ (yds)	0.125	0.289	0.135
$\bar{e}_\phi$ (deg)	-5.45×10^{-5}	6.59×10^{-5}	2.55×10^{-4}
$\sigma_{\bar{e}_\phi}$ (deg)	7.55×10^{-4}	2.37×10^{-3}	8.23×10^{-4}
$\bar{e}_\theta$ (deg)	1.48×10^{-4}	2.41×10^{-4}	1.94×10^{-4}
$\sigma_{\bar{e}_\theta}$ (deg)	7.43×10^{-4}	1.40×10^{-3}	7.34×10^{-4}

Table X

RATE ESTIMATION PERFORMANCE OF THE OPTIMIZED CORRELATED RANDOM ACCELERATION MODEL FILTERS

	FILTER		
PARAMETER	1A	2A	3A
$\bar{e}_{\dot{r}}$ (y/s)	3.77	3.83	3.78
$\sigma_{\bar{e}_{\dot{r}}}$ (y/s)	29.71	29.70	29.70
$\bar{e}_{\dot{\phi}}$ (d/s)	2.20×10^{-4}	1.64×10^{-2}	5.06×10^{-3}
$\sigma_{\bar{e}_{\dot{\phi}}}$ (d/s)	1.76×10^{-3}	0.109	1.39×10^{-2}
$\bar{e}_{\dot{\theta}}$ (d/s)	-1.58×10^{-4}	-1.50×10^{-4}	1.63×10^{-3}
$\sigma_{\bar{e}_{\dot{\theta}}}$ (d/s)	3.31×10^{-3}	8.14×10^{-3}	2.67×10^{-3}

Table XI

ACCELERATION ESTIMATION PERFORMANCE OF THE OPTIMIZED CORRELATED RANDOM ACCELERATION MODEL FILTERS

PARAMETER	FILTER 1A	2A	3A
$\bar{e}_r^{\cdot\cdot}$ (y/s^2)	-0.546	-0.549	-0.526
$\sigma_{\bar{e}_r^{\cdot\cdot}}$ (y/s^2)	3.32	3.39	3.17
$\bar{e}_\phi^{\cdot\cdot}$ (d/s^2)	-4.35×10^{-4}	1.82×10^{-2}	-1.12×10^{-3}
$\sigma_{\bar{e}_\phi^{\cdot\cdot}}$ (d/s^2)	2.68×10^{-3}	0.31	1.39×10^{-2}
$\bar{e}_\theta^{\cdot\cdot}$ (d/s^2)	-9.70×10^{-4}	0.49	5.39×10^{-4}
$\sigma_{\bar{e}_\theta^{\cdot\cdot}}$ (d/s^2)	1.52×10^{-3}	0.10	2.67×10^{-3}

IV. ADAPTIVE FILTERING

A. THE ADAPTIVE BANDWIDTH FILTER

A single fixed-parameter filter configuration cannot realistically be expected to perform well against all trajectory profiles likely to be encountered. One way of approaching this problem is to use a filter that adapts its characteristics to a changing tracking environment, e.g., a target maneuver is detected. The design problem now becomes two-fold; under what circumstances, or when, should the filter's characteristics be changed and by how much should the filter be changed? Price and Brown [Ref. 11] discuss some interesting approaches to the design of adaptive filters; unfortunately their complexity is beyond the defined scope of this thesis.

The single method to be considered here will be a simple switch-on-range scheme. The motivation for such a scheme lies in the simplicity of its implementation. Additionally, Parr reported [Ref. 4] that of the methods he investigated, the switch-on-range method provided the best overall performance in his simulations. When filter tracking is initiated, a low $\underset{\sim}{Q}(K)$ gain schedule (corresponding to a narrow bandwidth filter) is used to provide the optimum performance against long-range targets. When the target's measured range reaches a predetermined switching point, the filter switches to a high $\underset{\sim}{Q}(K)$ gain

schedule (corresponding to a wideband filter) for the short-range tracking situation.

Both filter models developed in Chapter II are simulated in a switch-on-range adaptive mode of operation. Again, to duplicate Parr's testing environment, a switching range of 3333.33 yards is used, and the filters are tested against trajectories 9 - 14. The profiles for trajectories 9 - 14 are identical to Parr's trajectories 1 - 6. The results reported for the adaptive simulations are the products of 80-member ensemble Monte Carlo simulations. For the simulations of the constant-velocity model, the low $\underset{\sim}{Q}(K)$ gain schedule used was that from Filter 3. The data in Table II show that Filter 3 gave the best overall performance against the long-range trajectories treated in Chapter III. Similar considerations yield Filter 4 as the source for the high $\underset{\sim}{Q}(K)$ gain schedule for the short-range tracking environment. When filter tracking is initiated, the Kalman gains used are those for Filter 3. When the target's measured range becomes less than or equal to the switching range of 3333.33 yards, the Kalman gain schedule is reset to the steady-state gain values obtained from the gain schedule for Filter 4. Similar consideration of the data presented in Table VII gives Filter 2A as the source of the high $\underset{\sim}{Q}(K)$ gain schedule while Filter 3A provides the low $\underset{\sim}{Q}(K)$ gain schedule. As in the above simulation, the low $\underset{\sim}{Q}(K)$ gain schedule is used until the

switching range (3333.33 yards) is reached. At that point, the Kalman gain schedule is reset to the steady-state gain values from gain schedule for Filter 2A. Initialization of the filters was the same for all simulations conducted and is described by Equations (2.5), (2.8), and (2.9). The statistics assumed in all simulations for the measurement noise process $\underset{\sim}{v}(K)$ are given by Equations (1.6) and (3.1).

B. SIMULATION RESULTS FOR THE CONSTANT-VELOCITY MODEL SWITCH-ON-RANGE ADAPTIVE FILTER

The constant-velocity model filter was tested in a switch-on-range adaptive mode of operation against trajectories 9 - 14. Table XII shows the filter's position prediction accuracy results. The data tabulated in Table XIII compare the results for the adaptive simulations of the models using the two different state vectors. Tables XIV and XV show the tracking performance of the filter in the adaptive mode. As in Chapter III the tabulated data represent the time-averaged means of estimation error and the RMS value of the deviation of the means of estimation error from their time average.

C. SIMULATION RESULTS FOR THE CORRELATED RANDOM ACCELERATION MODEL SWITCH-ON-RANGE ADAPTIVE FILTER

The correlated random acceleration model was also tested in an adaptive mode of operation against trajectories 9 - 14. Table XVI shows this filter's position prediction performance results. Tables XVII - XIX show the filter's

Table XII

SWITCH-ON-RANGE ADAPTIVE FILTER SIMULATION

FOR THE CONSTANT-VELOCITY MODEL

TRAJECTORY	%HIT	%SCARE
9	90.58	5.55
10	60.57	12.98
11	71.58	11.39
12	63.95	15.45
13	67.37	9.98
14	74.67	9.65

Table XIII

COMPARISON OF RESULTS FOR THE SPHERICAL AND CARTESIAN COORDINATE STATE VECTOR CONSTANT-VELOCITY MODEL SWITCH-ON-RANGE ADAPTIVE FILTERS

TRAJECTORY	STATE VECTOR			
	SPHERICAL		CARTESIAN	
	%HIT	%SCARE	%HIT	%SCARE
9	90.58	5.55	81.80	11.60
10	60.57	12.98	47.20	10.60
11	71.58	11.39	60.10	14.20
12	63.95	15.45	52.30	13.40
13	67.37	9.98	60.20	15.00
14	74.67	9.67	54.70	13.10

Table XIV

POSITION ESTIMATION PERFORMANCE FOR THE CONSTANT-VELOCITY MODEL

SWITCH-ON-RANGE ADAPTIVE FILTER

TRAJECTORY	$\bar{e}_r$(yds)	$\sigma_{\bar{e}_r}$(yds)	$\bar{e}_\phi$(deg)	$\sigma_{\bar{e}_\phi}$(deg)	$\bar{e}_\theta$(deg)	$\sigma_{\bar{e}_\theta}$(deg)
9	6.34×10^{-2}	0.105	1.31×10^{-4}	8.59×10^{-4}	-1.18×10^{-5}	9.01×10^{-4}
10	-2.82×10^{-3}	0.261	1.52×10^{-2}	6.52×10^{-2}	3.59×10^{-6}	9.16×10^{-4}
11	-3.30×10^{-2}	0.228	4.60×10^{-3}	3.46×10^{-2}	-4.79×10^{-5}	1.03×10^{-3}
12	-1.30×10^{-2}	0.220	1.12×10^{-4}	2.30×10^{-2}	-1.18×10^{-5}	9.30×10^{-4}
13	-0.131	0.478	1.14×10^{-4}	4.33×10^{-2}	-1.18×10^{-5}	9.75×10^{-4}
14	-0.167	0.509	6.14×10^{-3}	7.85×10^{-2}	-1.2×10^{-4}	1.75×10^{-3}

Table XV

RATE ESTIMATION PERFORMANCE FOR THE CONSTANT-VELOCITY MODEL SWITCH-ON-RANGE ADAPTIVE FILTER

TRAJECTORY	$\bar{e}_{\dot{r}}$(y/s)	$\sigma_{\bar{e}_{\dot{r}}}$(y/s)	$\bar{e}_{\dot{\phi}}$(d/s)	$\sigma_{\bar{e}_{\dot{\phi}}}$(d/s)	$\bar{e}_{\dot{\theta}}$(d/s)	$\sigma_{\bar{e}_{\dot{\theta}}}$(d/s)
9	1.27	15.81	6.94×10^{-5}	1.16×10^{-3}	-1.14×10^{-4}	1.50×10^{-3}
10	1.26	15.81	1.75×10^{-3}	9.92×10^{-2}	-1.46×10^{-4}	1.93×10^{-3}
11	1.24	15.81	1.96×10^{-2}	0.165	-1.24×10^{-3}	6.84×10^{-3}
12	1.25	15.81	-1.97×10^{-3}	0.109	-2.28×10^{-4}	2.40×10^{-3}
13	1.22	15.82	-6.48×10^{-3}	0.216	-1.87×10^{-4}	3.71×10^{-3}
14	1.18	15.82	1.47×10^{-2}	0.453	-1.92×10^{-3}	1.56×10^{-2}

tracking performance in the adaptive mode. As in the preceding section, the tabulated data represent the time-averaged means of estimation error and the RMS value of the deviation of the means of estimation error from their time average.

Table XVI

SWITCH-ON-RANGE ADAPTIVE FILTER SIMULATION

FOR THE CORRELATED RANDOM ACCELERATION MODEL

TRAJECTORY	%hit	%scare
9	62.27	10.85
10	46.72	15.20
11	60.46	10.76
12	53.34	13.06
13	56.67	9.36
14	56.08	11.23

Table XVII

POSITION ESTIMATION PERFORMANCE FOR THE CORRELATED RANDOM ACCELERATION SWITCH-ON-RANGE ADAPTIVE FILTER

TRAJECTORY	$\bar{e}_r$(yds)	$\sigma_{\bar{e}_r}$ (yds)	$\bar{e}_\phi$(deg)	$\sigma_{\bar{e}_\phi}$ (deg)	$\bar{e}_\theta$(deg)	$\sigma_{\bar{e}_\theta}$ (deg)
9	3.64×10^{-2}	0.12	1.57×10^{-4}	1.13×10^{-3}	1.81×10^{-4}	1.11×10^{-3}
10	1.01×10^{-2}	0.20	-1.42×10^{-2}	5.69×10^{-2}	1.96×10^{-4}	1.12×10^{-3}
11	-1.24×10^{-3}	0.16	1.66×10^{-4}	1.49×10^{-3}	1.82×10^{-4}	1.11×10^{-3}
12	1.57×10^{-2}	0.17	1.60×10^{-4}	1.84×10^{-3}	1.81×10^{-4}	1.11×10^{-3}
13	-2.10×10^{-2}	0.31	1.57×10^{-4}	3.98×10^{-3}	1.81×10^{-4}	1.11×10^{-3}
14	-7.32×10^{-2}	0.32	-1.58×10^{-5}	5.01×10^{-3}	1.85×10^{-4}	1.13×10^{-3}

Table XVIII

RATE ESTIMATION PERFORMANCE FOR THE CORRELATED RANDOM ACCELERATION MODEL SWITCH-ON-RANGE ADAPTIVE FILTER

TRAJECTORY	$\bar{e}_{\dot{r}}$ (y/s)	$\sigma_{\bar{e}_{\dot{r}}}$ (y/s)	$\bar{e}_{\dot{\phi}}$ (d/s)	$\sigma_{\bar{e}_{\dot{\phi}}}$ (d/s)	$\bar{e}_{\dot{\theta}}$ (d/s)	$\sigma_{\bar{e}_{\dot{\theta}}}$ (d/s)
9	1.26	15.81	2.16×10^{-4}	4.95×10^{-3}	2.87×10^{-4}	5.16×10^{-3}
10	1.26	15.82	2.70×10^{-3}	0.11	2.56×10^{-4}	5.32×10^{-3}
11	1.24	15.82	2.76×10^{-3}	4.44×10^{-2}	-7.30×10^{-4}	7.87×10^{-3}
12	1.25	15.82	-1.82×10^{-3}	2.03×10^{-2}	1.73×10^{-4}	5.36×10^{-3}
13	1.24	15.82	-6.35×10^{-3}	5.70×10^{-2}	2.14×10^{-4}	5.79×10^{-3}
14	1.20	15.84	-9.02×10^{-3}	0.20	-1.15×10^{-3}	1.15×10^{-2}

Table XIX

ACCELERATION ESTIMATION PERFORMANCE FOR THE CORRELATED RANDOM ACCELERATION MODEL SWITCH-ON-RANGE ADAPTIVE FILTER

TRAJECTORY	$\bar{\ddot{e}}_r$ (y/s^2)	$\sigma_{\bar{\ddot{e}}_r}$ (y/s^2)	$\bar{\ddot{e}}_\phi$ (d/s^2)	$\sigma_{\bar{\ddot{e}}_\phi}$ (d/s^2)	$\bar{\ddot{e}}_\theta$ (d/s^2)	$\sigma_{\bar{\ddot{e}}_\theta}$ (d/s^2)
9	-1.09	1.72	3.70×10^{-4}	1.23×10^{-2}	-4.94×10^{-2}	3.99×10^{-2}
10	-0.112	1.73	-3.50×10^{-3}	0.236	-4.84×10^{-3}	2.76
11	0.108	1.73	-3.07×10^{-2}	0.566	-0.266	3.37
12	-0.112	1.73	-1.01×10^{-2}	0.111	-1.44×10^{-2}	2.99
13	-0.119	1.81	-1.98×10^{-2}	0.263	7.91×10^{-2}	6.34

V. SUMMARY AND CONCLUSIONS

A. SUMMARY OF TEST RESULTS

1. Summary of Simulation Results for the Two Models

The results presented in Chapter III show that against non-maneuvering target profiles the optimized constant-velocity models provided superior prediction performance. When the bandwidth of the filter is increased, adequate position prediction performance against maneuvering targets is obtained but at the expense of performance degradation against the non-maneuvering profiles. The optimized correlated random acceleration model showed only a slight improvement against the maneuvering profiles considered, but produced results inferior to those obtained from the constant-velocity model against the non-maneuvering target profiles. The performance of the constant-velocity model against long-range maneuvering profiles was significantly lower than for the short-range maneuvering profiles. This is to be expected since the long-range profiles require that the predictions be made over a longer period of time during which the target may maneuver unexpectedly and significantly. The tabulated data presenting the tracking performance of the filters developed in Chapter II, show that the position estimation performance for both filters is approximately the same. The correlated random acceleration model gave only

slightly better rate estimation performance than did the constant-velocity model. Against the target profiles considered the filter models derived from a spherical coordinate state vector provided significantly better overall predicted position results than the filter models derived from a Cartesian Coordinate state vector.

2. Summary of Simulation Results for the Adaptive Filters

The results reported in Chapter IV for the simulation of the switch-on-range adaptive filters indicated that the constant-velocity model derived from a spherical coordinate state vector provided significantly better predicted position performance than the constant-velocity model filter derived from a Cartesian Coordinate state vector against all target profiles considered. Further, the results indicate that of the models derived from a spherical coordinate state vector, the constant-velocity model gave the best overall predicted position performance.

B. CONCLUSIONS

A filter derived from a spherical coordinate state vector with assumed linear system dynamics produces predicted position results potentially superior to those obtained from a filter derived from a Cartesian Coordinate state vector. The spherical coordinate state vector constant-velocity model provides better overall predicted

position results than the correlated random acceleration model derived from a spherical coordinate state vector. The reason for this is perhaps due to the manner in which the predicted position, $\underset{\sim}{\hat{x}}(K+N/K)$ is generated. For both models considered, the predicted position $\underset{\sim}{\hat{x}}(K+N/K)$ is formed by raising the state transition matrix $\underset{\sim}{\phi}(K,K-1)$, to be the $N^{\underline{th}}$ power as shown by Equation (C.2). In the case of the constant-velocity model such a method uses the current position and velocity estimates to generate the predicted position $\underset{\sim}{\hat{x}}(K+N/K)$. The correlated random acceleration model uses the current position, velocity and acceleration estimates to generate the predicted position. Thus, any error in the estimate of the target's acceleration state causes significant errors in the predicted position when propagated over the prediction interval. These acceleration induced prediction errors are not present when only the current estimates of position and velocity are used as in the case of the constant-velocity model.

A filter model derived from a spherical coordinate state vector can produce results potentially superior to those obtained from the corresponding Cartesian Coordinate state vector model in an adaptive filtering environment.

C. SUGGESTIONS FOR FURTHER INVESTIGATION

Since the constant-velocity model produced the best overall predicted position results, a complete analysis of the sensitivity of this model's predicted position

performance to the statistics assumed for the measurement noise process $\underset{\sim}{v}(K)$ and the sampling period T should be carried out. Additional sensitivity studies should also focus on allowing the diagonal elements of $\underset{\sim}{Q}(K)$ to vary independently, thus removing the constraint that $q_{22} = q_{33}$.

For the correlated random acceleration model, one area that warrants future investigation is the use of only the position and velocity estimates of the filter in generating the predicted position, and comparing the results with those obtained from the constant-velocity model filter. This would reduce the errors in the predicted position due to errors in the current estimate of the target's acceleration state.

Both models should be tested against a greater variety of target profiles, especially targets that present crossing profiles in which large bearing and elevation rates are generated, and targets that execute maneuvers while in level flight.

Finally, more sophisticated adaptive filtering schemes should be developed and investigated for the constant-velocity model. A prime candidate for such an investigation is the residual testing method reported by Price and Brown [Ref. 11].

APPENDIX A

MONTE CARLO SIMULATION PROGRAM

A.1 PROGRAM LISTING AND DESCRIPTION

The Monte Carlo Simulation Program used in this thesis was written by Dr. D. E. Kirk of the Naval Postgraduate School. It is extremely flexible and well documented to assist the potential user in data preparation and input, as well as the various options available. It should be pointed out that SUBROUTINE KILL and all statements in the main program that deal with miss distance data were written specifically for this thesis. They may be deleted with no loss of program flexibility or functional operation.

Computer system requirements vary with the simulation and the user's desires. All the computer work done for this thesis was performed on the IBM SYSTEM 360/67 at the Naval Postgraduate School. The program as currently dimensioned requires 520K bytes of storage. When redimensioned for the 60 data point trajectories, 320K bytes of storage are required. A 100-member ensemble simulation of a linear sixth-order filter with no on-line computation required, the trajectory read in, miss distance information desired, and all printing and plotting options invoked requires approximately 1.5 minutes of CPU time for execution. An additional 30 seconds (approximately) are required for the compile/link process. For the work done for this thesis

the program was pre-compiled (FORTRAN IV Level G) and stored and executed from a data cell. The link process in this case required approximately two seconds. If the same linear, sixth-order filter is again simulated but the printed and plotted output is suppressed (retaining only the miss distance data) 59 seconds of CPU time is required to execute the simulation. In the case of the linear ninth-order filters that were simulated, the storage requirements remain unchanged, as do the time requirements for the compile/link process. A 100-member ensemble simulation of the linear ninth-order filter with no on-line computation required, miss distance information desired, the trajectory read in and all printing and plotting options invoked requires approximately 3.0 minutes of CPU time for execution. If only the miss distance information is desired and all other output is suppressed, approximately 2.35 minutes of CPU time is required to execute the simulation.

```
C      THIS PROGRAM PERFORMS MONTE CARLO SIMULATION OF STATE ESTIMATORS  MCSP    1
C      OF WHICH THE KALMAN FILTER IS ONE EXAMPLE. THERE ARE SEVERAL      MCSP    2
C      OPTIONS AVAILABLE AS INDICATED IN THE DETAILED COMMENTS BELOW.    MCSP    3
C      IT SHOULD BE NOTED THAT ALL COMPUTATIONS OF GAINS USING THE       MCSP    4
C      SUBROUTINE GAIN ARE PERFORMED IN DOUBLE PRECISION. THUS ALL       MCSP    5
C      ARRAYS FOR USE IN "GAIN" MUST BE PREPARED ACCORDINGLY.            MCSP    6
C                                                                        MCSP    7
       REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI,PKKM10MCSP    8
      1,MISSD                                                           MCSP    9
       COMMON GAMMA(9,9),COVW(9,9),R(9,9),PHI(9,9),H(9,9),TEMP(9,9),TEMP1MCSP   10
      1(9,9),TEMP2(9,9),PKKM1(9,9),PKKM10(9,9),G(9,9),PKK(9,9),Q(9,9),EI(MCSP   11
      29,9),MISSD(100,160),VAR(9,9,160),GKS(9,9,160),PKKS(9,9,160),XM(9,1MCSP   12
      360),ERR(9,160),GAMMAS(9,9),PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGMCSP   13
      4W(9),X(9),SIGXZ(9),XZMEAN(9),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SMCSP   14
      5IGV(9),XHATZ(9),XINT(160),N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXMCSP   15
      6Z,IV,IW,IEST,ND,NHIT,NSCARE,NMISS,NFIRE                          MCSP   16
       DIMENSION XP(160), YP(160), SMISS(100,160), TEMPA(160)           MCSP   17
C                                                                       MCSP   18
C                                                                       MCSP   19
C      N=ORDER OF SYSTEM MODEL AND FILTER (DIMENSION OF X,XHAT)         MCSP   20
C                                                                       MCSP   21
C      M=NUMBER OF MEASUREMENTS (DIMENSION OF THE VECTOR Z)             MCSP   22
C                                                                       MCSP   23
C      IN=NUMBER OF INPUT RANDOM FORCING FCNS (=DIMENSION OF W)         MCSP   24
C                                                                       MCSP   25
C      NSAM=NUMBER OF TIME SAMPLES                                      MCSP   26
C                                                                       MCSP   27
C      NENS=NUMBER OF MEMBERS IN ENSEMBLE                               MCSP   28
C                                                                       MCSP   29
       READ (5,106) N,M,IN,NSAM,NENS                                    MCSP   30
C                                                                       MCSP   31
C                                                                       MCSP   32
       READ (5,107) ND                                                  MCSP   33
C      THE VALUE OF ND READ IN MUST EQUAL THE ROW (AND COLUMN) DIMENSION MCSP   34
C      SPECIFIED FOR THE SQUARE MATRIX "TEMP1", E.G. IF TEMP1(3,3) IS   MCSP   35
C      SPECIFIED IN THE COMMON STATEMENT "ND" MUST BE EQUAL TO 3.        MCSP   36
C                                                                       MCSP   37
C                                                                       MCSP   38
C      IG=-1 -- GAINS COMPUTED OFF-LINE AND READ IN                     MCSP   39
C          0 -- GAINS COMPUTED ONLY ONCE BEFORE STARTING MONTE CARLO    MCSP   40
C          1 -- GAINS COMPUTED FOR EACH MEMBER OF ENSEMBLE              MCSP   41
C                                                                       MCSP   42
C      IFLR=0 -- R IS READ IN                                           MCSP   43
C      IFLR.NE.0 -- R IS COMPUTED ON-LINE AT EACH TIME SAMPLE           MCSP   44
C                                                                       MCSP   45
C      IEST=0 -- STANDARD KALMAN FILTER EQUATION IS USED                MCSP   46
C      IEST.NE.0 -- STD. KALMAN FILTER EQ. NOT USED                     MCSP   47
C                                                                       MCSP   48
```

```
C      ITRK= 0 -- SEVERAL TRACKS GENERATED FROM STD LINEAR EQS                MCSP  49
C      ITRK= 1 -- ONLY ONE TRACK IS USED                                      MCSP  50
C      ITRK NE 0 OR 1 -- SEVERAL TRACKS GENERATED BUT NOT FROM STANDARD       MCSP  51
C                        LINEAR DIFFERENCE EQUATIONS.                         MCSP  52
C                                                                             MCSP  53
C      ISTAT=0 -- MEAN OF TRACK,MEAN & VARIANCE OF EST. ERROR COMPUTED        MCSP  54
C      ISTAT.NE.0 -- SAME AS ISTAT=0 BUT OFF-DIAGONAL TERMS IN COVARIANCE     MCSP  55
C                                                                             MCSP  56
C                    MATRIX ARE ALSO COMPUTED                                 MCSP  57
C                                                                             MCSP  58
      READ (5,108) IG,IFLR,IEST,ITRK,ISTAT,IQ,ITRC                           MCSP  59
C                                                                             MCSP  60
C     IG=-1 -- Q IS COMPUTED ON-LINE AT EACH SAMPLE BY "QON"                  MCSP  61
C         0 -- Q MATRIX IS READ IN                                            MCSP  62
C         1 -- COVARIANCE OF W READ IN AND Q IS  COMPUTED ONCE BEFORE         MCSP  63
C              MONTE CARLO BEGINS BY CALLING "QMAT"                           MCSP  64
C                                                                             MCSP  65
C     ITRO=0 -- ONLY USED IF ONE TRACK; ITRO=0 INDICATES THE                  MCSP  66
C               TRACK IS READ IN (S.P.), OTHERWISE THE TRACK                  MCSP  67
C               IS GENERATED BY A SINGLE CALL OF SUBROUTINE                   MCSP  68
C               TRACK.                                                        MCSP  69
C                                                                             MCSP  70
C     IPRT=0 -- SOME OR ALL OUTPUT DATA IS PRINTED                            MCSP  71
C                                                                             MCSP  72
C     IPLT=0 -- SOME OR ALL OUTPUT DATA IS PLOTTED                            MCSP  73
      READ (5,109) IPRT,IPLT                                                  MCSP  74
C                                                                             MCSP  75
C     IGPLT,ITHVPL,IMTPLT,ISMPLT,ISVPLT ARE PLOTTING INDICATORS               MCSP  76
C     IF THE INDICATOR IS EQUAL TO 1 PLOTS ARE GENERATED. THE INDICATORS      MCSP  77
C     CORRESPOND TO THE FOLLOWING QUANTITIES                                  MCSP  78
C     IGPLT(GAIN VS K),ITHVPL (THEORETICAL COVARIANCE MATRIX VS K),           MCSP  79
C     IMTPLT (SAMPLE MEAN OF TRACK=TRACK IF ONLY ONE TRACK),                  MCSP  80
C     ISMPLT (SAMPLE MEAN OF EST. ERROR VS K), ISVPLT (SAMPLE VARIANCE        MCSP  81
C     OF EST. ERROR VS K)                                                     MCSP  82
      IF (IPLT.EQ.0) READ (5,110) IGPLT,ITHVPL,IMTPLT,ISMPLT,ISVPLT          MCSP  83
C                                                                             MCSP  84
C     IPHI=0 -- PHI MATRIX TO BE READ IN (IN DOUBLE PREC)                     MCSP  85
C     IPHI.NE.0 -- PHI MATRIX NOT TO BE READ                                  MCSP  86
C                                                                             MCSP  87
C     IH=0 -- H MATRIX TO BE READ IN (IN D.P.)                                MCSP  88
C                                                                             MCSP  89
C     IR=0 -- R MATRIX IS READ IN                                             MCSP  90
C                                                                             MCSP  91
C     IPKKM0=0 -- THE STARTING VALUE FOR THE MATRIX P(K/K-1) IS TO            MCSP  92
C                 BE READ IN (D.P.)                                           MCSP  93
C                                                                             MCSP  94
C     IGAM=0 -- GAMMA MATRIX TO BE READ IN (D.P.)                             MCSP  95
C                                                                             MCSP  96
```

```
C      ISIGV=0 -- STD. DEVIATIONS OF MEASUREMENT NOISE TO BE READ IN (SP)MCSP  97
C                                                                       MCSP  98
C      ISIGW=0 -- STD. DEVIATIONS OF RANDOM INPUTS TO BE READ IN (SP)   MCSP  99
C                                                                       MCSP 100
C      IXHZ=0 -- INITIAL VALUE OF THE PREDICTED VALUE XH(0/-1) TO       MCSP 101
C                BE READ IN (S.P.)                                      MCSP 102
C                                                                       MCSP 103
C      IC=1 -- ONE INITIAL CONDITION VALUE USED FOR THE STATE           MCSP 104
C              OTHERWISE THE VALUE OF X(0) IS GENERATED BY A RANDOM     MCSP 105
C              NUMBER GENERATOR CALLED BY "XZERO".                      MCSP 106
C                                                                       MCSP 107
      READ (5,111) IPHI,IH,IR,IPKKMO,IGAM,ISIGV,ISIGW,IXHZ,IC,IMISS,IFIRMCSP 108
     1E,IMISSK,IMISSR,IMISSI,IMISSC                                     MCSP 109
C                                                                       MCSP 110
C      IMISS=0        FIRING AND MISS DISTANCE DATA ARE DESIRED.  OTHER-MCSP 111
C                     WISE NOT.                                         MCSP 112
C                                                                       MCSP 113
C      IFIRE=0        FIRING DATA AVERAGED OVER ENTIRE MONTE CARLO RUN IS MCSP 114
C                     PRINTED.  OTHERWISE NOT.                          MCSP 115
C                                                                       MCSP 116
C      IMISSK=0       MEAN AND STANDARD DEVIATION OF THE MISS DISTANCE AREMCSP 117
C                     PLOTTED VS. K.  OTHERWISE NOT.                    MCSP 118
C                                                                       MCSP 119
C      IMISSR=0       MEAN AND STANDARD DEVIATION OF THE MISS DISTANCE AREMCSP 120
C                     PLOTTED VS. THE TARGET SLANT RANGE.  OTHERWISE NOT. MCSP 121
C                                                                       MCSP 122
C      IMISSI=0       MEAN AND STANDARD DEVIATION OF THE MISS DISTANCE AREMCSP 123
C                     PLOTTED VS. THE AVERAGE LENGTH OF THE PREDICTION  MCSP 124
C                     INTERVAL.  OTHERWISE NOT.                         MCSP 125
C                                                                       MCSP 126
C      IMISSC=0       MEANS AND STANDARD DEVIATIONS OF THE MISS DISTANCE MCSP 127
C                     ARE PRINTED OUT.  OTHERWISE NOT.                  MCSP 128
C                                                                       MCSP 129
      NHIT = 0                                                          MCSP 130
      NMISS = 0                                                         MCSP 131
      NSCARE = 0                                                        MCSP 132
      IRDS = 0                                                          MCSP 133
      NFIRE = 0                                                         MCSP 134
      IMISSN = 0                                                        MCSP 135
      ISCARE = 0                                                        MCSP 136
      CALL OVFLOW                                                       MCSP 137
      IW = 6395217                                                      MCSP 138
      IV = 1936748                                                      MCSP 139
      IXZ = 135769                                                      MCSP 140
C                                                                       MCSP 141
CC                                                                      MCSP 142
C     THE FOLLOWING SECTION PRINTS OUT A DESCRIPTION OF THE RUN AS      MCSP 143
C     SPECIFIED BY THE USER'S FLAGS                                     MCSP 144
```

```
C                                                                       MCSP  145
C                                                                       MCSP  146
      WRITE (6,181)                                                     MCSP  147
      WRITE (6,112)                                                     MCSP  148
      IF (IG.EQ.0) GO TO 1                                              MCSP  149
      IF (IG.EQ.1) GO TO 2                                              MCSP  150
      WRITE (6,113)                                                     MCSP  151
      GO TO 3                                                           MCSP  152
    1 WRITE (6,114)                                                     MCSP  153
      GO TO 3                                                           MCSP  154
    2 WRITE (6,115)                                                     MCSP  155
    3 IF (IEST.EQ.0) GO TO 4                                            MCSP  156
      WRITE (6,116)                                                     MCSP  157
      GO TO 5                                                           MCSP  158
    4 WRITE (6,117)                                                     MCSP  159
    5 IF (ITRK.EQ.0) GO TO 8                                            MCSP  160
      IF (ITRK.EQ.-1) GO TO 7                                           MCSP  161
      IF (ITRO.EQ.0) GO TO 6                                            MCSP  162
      WRITE (6,118)                                                     MCSP  163
      GO TO 9                                                           MCSP  164
    6 WRITE (6,119)                                                     MCSP  165
      GO TO 9                                                           MCSP  166
    7 WRITE (6,120)                                                     MCSP  167
      GO TO 9                                                           MCSP  168
    8 WRITE (6,121)                                                     MCSP  169
    9 IF (ISTAT.EQ.0) GO TO 10                                          MCSP  170
      WRITE (6,122)                                                     MCSP  171
      GO TO 11                                                          MCSP  172
   10 WRITE (6,123)                                                     MCSP  173
   11 IF (IQ.EQ.0) GO TO 13                                             MCSP  174
      IF (IQ.EQ.1) GO TO 12                                             MCSP  175
      WRITE (6,124)                                                     MCSP  176
      GO TO 14                                                          MCSP  177
   12 WRITE (6,125)                                                     MCSP  178
      GO TO 14                                                          MCSP  179
   13 WRITE (6,126)                                                     MCSP  180
   14 IF (IFLR.EQ.0) GO TO 15                                           MCSP  181
      WRITE (6,127)                                                     MCSP  182
      GO TO 16                                                          MCSP  183
   15 WRITE (6,128)                                                     MCSP  184
   16 WRITE (6,129)                                                     MCSP  185
      IF (IPHI.EQ.0) WRITE (6,130)                                      MCSP  186
      IF (IH.EQ.0) WRITE (6,131)                                        MCSP  187
      IF (IFLR.EQ.0) WRITE (6,132)                                      MCSP  188
      IF (IQ.EQ.0) GO TO 17                                             MCSP  189
      IF (IQ.EQ.-1) GO TO 18                                            MCSP  190
      WRITE (6,133)                                                     MCSP  191
      GO TO 18                                                          MCSP  192
```

```
   17 WRITE (6,134)                                                    MCSP  193
   18 IF (IGAM.EQ.0) WRITE (6,135)                                     MCSP  194
      IF (ISIGV.EQ.0) WRITE (6,136)                                    MCSP  195
      IF (ISIGW.EQ.0) WRITE (6,137)                                    MCSP  196
      IF (IXHZ.EQ.0) WRITE (6,138)                                     MCSP  197
      IF (IC.NE.1) WRITE (6,139)                                       MCSP  198
      IF (IPKKM1.EQ.0) WRITE (6,140)                                   MCSP  199
C                                                                      MCSP  200
C                                                                      MCSP  201
C     THE FOLLOWING SECTION PRINTS OUT A DESCRIPTION OF THE OUTPUT     MCSP  202
C     DATA CALLED FOR                                                  MCSP  203
C                                                                      MCSP  204
C                                                                      MCSP  205
      WRITE (6,141)                                                    MCSP  206
      IF (IPRT.NE.0) GO TO 19                                          MCSP  207
      WRITE (6,142)                                                    MCSP  208
      IF (IG.EQ.0) WRITE (6,143)                                       MCSP  209
      WRITE (6,144)                                                    MCSP  210
      IF (ISTAT.NE.0) WRITE (6,145)                                    MCSP  211
      GO TO 20                                                         MCSP  212
   19 WRITE (6,146)                                                    MCSP  213
   20 IF (IMISS.NE.0) GO TO 21                                         MCSP  214
      IF (IFIRE.EQ.0) WRITE (6,192)                                    MCSP  215
      IF (IMISSC.EQ.0) WRITE (6,193)                                   MCSP  216
      GO TO 22                                                         MCSP  217
   21 WRITE (6,194)                                                    MCSP  218
   22 IF (IPLT.NE.0) GO TO 23                                          MCSP  219
      WRITE (6,147)                                                    MCSP  220
      IF (IGPLT.EQ.1) WRITE (6,148)                                    MCSP  221
      IF (ITHVPL.EQ.1) WRITE (6,149)                                   MCSP  222
      IF (IMTPLT.EQ.1) WRITE (6,150)                                   MCSP  223
      IF (ISMPLT.EQ.1) WRITE (6,151)                                   MCSP  224
      IF (ISVPLT.EQ.1) WRITE (6,152)                                   MCSP  225
      GO TO 24                                                         MCSP  226
   23 WRITE (6,153)                                                    MCSP  227
   24 CONTINUE                                                         MCSP  228
      IF (IMISS.NE.0) GO TO 25                                         MCSP  229
      WRITE (6,208)                                                    MCSP  230
      IF (IMISSK.EQ.0) WRITE (6,189)                                   MCSP  231
      IF (IMISSR.EQ.0) WRITE (6,190)                                   MCSP  232
      IF (IMISSI.EQ.0) WRITE (6,191)                                   MCSP  233
      GO TO 26                                                         MCSP  234
   25 WRITE (6,209)                                                    MCSP  235
   26 CONTINUE                                                         MCSP  236
      WRITE (6,181)                                                    MCSP  237
      WRITE (6,154)                                                    MCSP  238
      WRITE (6,155) N,M,IN,NSAM,NENS,ND                                MCSP  239
C                                                                      MCSP  240
```

```
C                                                                       MCSP  241
C     THE FOLLOWING SECTION READS THE SPECIFIED INPUT MATRICES          MCSP  242
C                                                                       MCSP  243
      IF (IPHI.NE.0) GO TO 28                                           MCSP  244
      CALL MREAD (PHI,N,N)                                              MCSP  245
C                                                                       MCSP  246
      DO 27 I=1,N                                                       MCSP  247
C                                                                       MCSP  248
      DO 27 J=1,N                                                       MCSP  249
   27 PHIS(I,J) = PHI(I,J)                                              MCSP  250
C                                                                       MCSP  251
      WRITE (6,156)                                                     MCSP  252
      CALL MWRITE (PHI,N,N)                                             MCSP  253
C                                                                       MCSP  254
C                                                                       MCSP  255
   28 IF (IH.NE.0) GO TO 30                                             MCSP  256
      CALL MREAD (H,M,N)                                                MCSP  257
C                                                                       MCSP  258
      DO 29 I=1,M                                                       MCSP  259
C                                                                       MCSP  260
      DO 29 J=1,N                                                       MCSP  261
   29 HS(I,J) = H(I,J)                                                  MCSP  262
C                                                                       MCSP  263
      WRITE (6,157)                                                     MCSP  264
      CALL MWRITE (H,M,N)                                               MCSP  265
C                                                                       MCSP  266
C                                                                       MCSP  267
   30 IF (IR.NE.0) GO TO 31                                             MCSP  268
      CALL MREAD (R,M,M)                                                MCSP  269
      WRITE (6,158)                                                     MCSP  270
      CALL MWRITE (R,M,M)                                               MCSP  271
C                                                                       MCSP  272
C                                                                       MCSP  273
   31 IF (IQ.NE.1) GO TO 32                                             MCSP  274
      CALL MREAD (COVW,IN,IN)                                           MCSP  275
      WRITE (6,159)                                                     MCSP  276
      CALL MWRITE (COVW,IN,IN)                                          MCSP  277
      GO TO 33                                                          MCSP  278
   32 IF (IQ.NE.0) GO TO 33                                             MCSP  279
      CALL MREAD (Q,N,N)                                                MCSP  280
      WRITE (6,160)                                                     MCSP  281
      CALL MWRITE (Q,N,N)                                               MCSP  282
C                                                                       MCSP  283
C                                                                       MCSP  284
   33 IF (IGAM.NE.0) GO TO 35                                           MCSP  285
      CALL MREAD (GAMMA,N,IN)                                           MCSP  286
C                                                                       MCSP  287
      DO 34 I=1,N                                                       MCSP  288
```

```
C                                                                       MCSP  289
      DO 34 J=1,IN                                                      MCSP  290
   34 GAMMAS(I,J) = GAMMA(I,J)                                          MCSP  291
C                                                                       MCSP  292
CC                                                                      MCSP  293
C                                                                       MCSP  294
      WRITE (6,161)                                                     MCSP  295
      CALL MWRITE (GAMMA,N,IN)                                          MCSP  296
C                                                                       MCSP  297
C                                                                       MCSP  298
   35 IF (IPKKMO.NE.0) GO TO 36                                         MCSP  299
      CALL MREAD (PKKM10,N,N)                                           MCSP  300
      WRITE (6,162)                                                     MCSP  301
      CALL MWRITE (PKKM10,N,N)                                          MCSP  302
C                                                                       MCSP  303
C                                                                       MCSP  304
   36 IF (ISIGV.NE.0) GO TO 37                                          MCSP  305
      CALL VREAD (SIGV,M)                                               MCSP  306
      WRITE (6,163)                                                     MCSP  307
      CALL VWRITE (SIGV,M)                                              MCSP  308
C                                                                       MCSP  309
C                                                                       MCSP  310
   37 IF (ISIGW.NE.0) GO TO 38                                          MCSP  311
      CALL VREAD (SIGW,IN)                                              MCSP  312
      WRITE (6,164)                                                     MCSP  313
      CALL VWRITE (SIGW,IN)                                             MCSP  314
C                                                                       MCSP  315
C                                                                       MCSP  316
   38 IF (IXHZ.NE.0) GO TO 39                                           MCSP  317
      CALL VREAD (XHATZ,N)                                              MCSP  318
      WRITE (6,165)                                                     MCSP  319
      CALL VWRITE (XHATZ,N)                                             MCSP  320
C                                                                       MCSP  321
C                                                                       MCSP  322
   39 IF (IC.EQ.1) GO TO 40                                             MCSP  323
C     IC.NE.1 MEANS THAT MEANS AND STD. DEVIATIONS OF THE INITIAL STATE MCSP  324
C     VALUE MUST BE READ IN. OTHERWISE NOT READ IN.                     MCSP  325
      CALL VREAD (XZMEAN,N)                                             MCSP  326
      WRITE (6,166)                                                     MCSP  327
      CALL VWRITE (XZMEAN,N)                                            MCSP  328
C                                                                       MCSP  329
C                                                                       MCSP  330
      CALL VREAD (SIGXZ,N)                                              MCSP  331
      WRITE (6,167)                                                     MCSP  332
      CALL VWRITE (SIGXZ,N)                                             MCSP  333
      GO TO 45                                                          MCSP  334
C                                                                       MCSP  335
C                                                                       MCSP  336
```

```
   40 READ (5,169) (XS(I,1),I=1,9)                                    MCSP  337
C     INITIAL CONDITION HAS BEEN READ                                 MCSP  338
      IF (ITRK.NE.1) GO TO 45                                         MCSP  339
      IF (ITRC.NE.0) GO TO 43                                         MCSP  340
C                                                                     MCSP  341
      DO 41 K=2,NSAM                                                  MCSP  342
   41 READ (5,169) (XS(I,K),I=1,9)                                    MCSP  343
C                                                                     MCSP  344
      IF (N.GT.6) GO TO 44                                            MCSP  345
C                                                                     MCSP  346
C                                                                     MCSP  347
      DO 42 K=1,NSAM                                                  MCSP  348
      XS(3,K) = XS(4,K)                                               MCSP  349
      XS(4,K) = XS(5,K)                                               MCSP  350
      XS(5,K) = XS(7,K)                                               MCSP  351
   42 XS(6,K) = XS(8,K)                                               MCSP  352
C                                                                     MCSP  353
CC                                                                    MCSP  354
C                                                                     MCSP  355
      WRITE (6,168)                                                   MCSP  356
      WRITE (6,171) (XS(I,1),I=1,N)                                   MCSP  357
      GO TO 44                                                        MCSP  358
   43 CALL TRACK                                                      MCSP  359
C     IF TRACK CALLED HERE IT SHOULD BE WRITTEN TO GENERATE AND       MCSP  360
C     STORE THE TRACK IN XS(N,K) FOR K=2(NOTE),NSAM                   MCSP  361
C                                                                     MCSP  362
   44 WRITE (6,170)                                                   MCSP  363
      WRITE (6,171) (XS(I,1),I=1,N)                                   MCSP  364
      WRITE (6,171) (XS(I,NSAM),I=1,N)                                MCSP  365
   45 CONTINUE                                                        MCSP  366
C                                                                     MCSP  367
CC                                                                    MCSP  368
CC    THE FOLLOWING SECTION PREPARES FOR THE MONTE CARLO LOOP         MCSP  369
CC                                                                    MCSP  370
CC    FORM NXN IDENTITY MATRIX IN DOUBLE PRECISION                    MCSP  371
CC                                                                    MCSP  372
C                                                                     MCSP  373
      DO 46 I=1,N                                                     MCSP  374
C                                                                     MCSP  375
      DO 46 J=1,N                                                     MCSP  376
      EI(I,J) = 0.D0                                                  MCSP  377
   46 IF (I.EQ.J) EI(I,J)=1.D0                                        MCSP  378
C                                                                     MCSP  379
CC                                                                    MCSP  380
CC    GIVEN THE MATRIX GAMMA AND THE COVARIANCE OF W COMPUTE Q        MCSP  381
C     USING DOUBLE PRECISION ARITHMETIC                               MCSP  382
      IF (IQ.NE.1) GO TO 47                                           MCSP  383
      CALL QMAT                                                       MCSP  384
```

```
      WRITE (6,160)                                                     MCSP  385
      CALL MWRITE (Q,N,N)                                               MCSP  386
   47 IF (IG.NE.-1) GO TO 49                                            MCSP  387
C                                                                       MCSP  388
C                                                                       MCSP  389
      DO 48 K=1,NSAM                                                    MCSP  390
C                                                                       MCSP  391
      DO 48 I=1,N                                                       MCSP  392
   48 READ (5,169) (GKS(I,J,K),J=1,M)                                   MCSP  393
C                                                                       MCSP  394
      GO TO 52                                                          MCSP  395
   49 IF (IG.NE.0) GO TO 52                                             MCSP  396
C                                                                       MCSP  397
C                                                                       MCSP  398
      DO 51 K=1,NSAM                                                    MCSP  399
      CALL GAIN                                                         MCSP  400
C                                                                       MCSP  401
C                                                                       MCSP  402
      DO 51 I=1,N                                                       MCSP  403
C                                                                       MCSP  404
      DO 50 L=1,N                                                       MCSP  405
   50 PKKS(I,L,K) = PKK(I,L)                                            MCSP  406
C                                                                       MCSP  407
C                                                                       MCSP  408
      DO 51 J=1,M                                                       MCSP  409
   51 GKS(I,J,K) = G(I,J)                                               MCSP  410
C                                                                       MCSP  411
   52 CONTINUE                                                          MCSP  412
C     IF GAINS WERE TO BE READ IN (IG=-1) OR COMPUTED ONLY              MCSP  413
C     ONCE(IG=0), THIS HAS NOW BEEN DONE                                MCSP  414
C                                                                       MCSP  415
C                                                                       MCSP  416
C     SET UP ARRAYS FOR COMPUTING STATISTICS                            MCSP  417
C                                                                       MCSP  418
C                                                                       MCSP  419
      DO 54 I=1,NENS                                                    MCSP  420
C                                                                       MCSP  421
      DO 53 J=1,NSAM                                                    MCSP  422
   53 MISSD(I,J) = 0.0D0                                                MCSP  423
C                                                                       MCSP  424
   54 CONTINUE                                                          MCSP  425
C                                                                       MCSP  426
C                                                                       MCSP  427
      DO 55 K=1,NSAM                                                    MCSP  428
      YP(K) = 0.0                                                       MCSP  429
      XP(K) = 0.0                                                       MCSP  430
      XINT(K) = 0.0                                                     MCSP  431
      TEMPA(K) = 0.0                                                    MCSP  432
```

```
C                                                                       MCSP  433
      DO 55 J=1,N                                                       MCSP  434
      XM(J,K) = 0.                                                      MCSP  435
      ERR(J,K) = 0.                                                     MCSP  436
C                                                                       MCSP  437
      DO 55 L=1,N                                                       MCSP  438
   55 VAR(J,L,K) = 0.                                                   MCSP  439
C                                                                       MCSP  440
CC                                                                      MCSP  441
CC                                                                      MCSP  442
CC    BEGIN MAIN ITERATION LOOP HERE                                    MCSP  443
CC                                                                      MCSP  444
CC                                                                      MCSP  445
C                                                                       MCSP  446
      DO 62 ITER=1,NENS                                                 MCSP  447
      IF (IC.EQ.1) GO TO 56                                             MCSP  448
      CALL XZERO                                                        MCSP  449
C                                                                       MCSP  450
   56 DO 57 I=1,N                                                       MCSP  451
   57 XHKKM1(I) = XHATZ(I)                                              MCSP  452
C                                                                       MCSP  453
CC                                                                      MCSP  454
      DO 61 K=1,NSAM                                                    MCSP  455
C                                                                       MCSP  456
CC    FORM NOISY MEASUREMENT FROM TRUE STATE VALUE                      MCSP  457
CC                                                                      MCSP  458
C                                                                       MCSP  459
      DO 58 I=1,N                                                       MCSP  460
   58 X(I) = XS(I,K)                                                    MCSP  461
C                                                                       MCSP  462
CC                                                                      MCSP  463
      CALL MEAS                                                         MCSP  464
C     GAIN IS NOT TO BE COMPUTED ON-LINE IF IG.NE.1                     MCSP  465
      IF (IG.NE.1) GO TO 60                                             MCSP  466
C                                                                       MCSP  467
CC    Q IS TO BE COMPUTED ON-LINE IF IFLQ.NE.0                          MCSP  468
      IF (IFLQ.NE.0) CALL QCN                                           MCSP  469
C     R IS TO BE COMPUTED ON-LINE IF IFLR.NE.0                          MCSP  470
CC                                                                      MCSP  471
      IF (IFLR.NE.0) CALL RCN                                           MCSP  472
      CALL GAIN                                                         MCSP  473
C                                                                       MCSP  474
CC                                                                      MCSP  475
      DO 59 I=1,N                                                       MCSP  476
C                                                                       MCSP  477
      DO 59 J=1,M                                                       MCSP  478
   59 GKS(I,J,K) = G(I,J)                                               MCSP  479
C                                                                       MCSP  480
```

```
C                                                                       MCSP  481
C     UPDATE THE STATE ESTIMATE                                         MCSP  482
C                                                                       MCSP  483
   60 CALL ESTIM                                                        MCSP  484
C                                                                       MCSP  485
C     UPDATE RUNNING SUMS USED IN COMPUTING STATISTICS                  MCSP  486
      CALL STAT                                                         MCSP  487
      IF (IMISS.NE.1) CALL KILL                                         MCSP  488
      IF (K.EQ.NSAM) GO TO 62                                           MCSP  489
C                                                                       MCSP  490
C     UPDATE TRACK BY COMPUTING X(K+1)                                  MCSP  491
      IF (ITRK.NE.1) CALL TRACK                                         MCSP  492
   61 CONTINUE                                                          MCSP  493
C                                                                       MCSP  494
C                                                                       MCSP  495
   62 CONTINUE                                                          MCSP  496
C                                                                       MCSP  497
CC                                                                      MCSP  498
C     DIVIDE RUNNING SUMS COMPUTED BY SUBROUTINE STAT BY ENSEMBLE       MCSP  499
C     SIZE TO COMPUTE STATISTICS                                        MCSP  500
CC                                                                      MCSP  501
C                                                                       MCSP  502
      ENS = NENS                                                        MCSP  503
C                                                                       MCSP  504
      DO 64 K=1,NSAM                                                    MCSP  505
C                                                                       MCSP  506
      DO 64 J=1,N                                                       MCSP  507
      IF (ITRK.EQ.1) GO TO 63                                           MCSP  508
      XM(J,K) = XM(J,K)/ENS                                             MCSP  509
   63 ERR(J,K) = ERR(J,K)/ENS                                           MCSP  510
   64 VAR(J,J,K) = VAR(J,J,K)/ENS-ERR(J,K)**2                           MCSP  511
C                                                                       MCSP  512
CC                                                                      MCSP  513
C                                                                       MCSP  514
      IF (ISTAT.EQ.0) GO TO 66                                          MCSP  515
C                                                                       MCSP  516
C     COMPUTE OFF-DIAGONAL TERMS IN COVARIANCE OF ESTIMATION            MCSP  517
C     ERROR MATRIX IF ISTAT.NE.0                                        MCSP  518
CC                                                                      MCSP  519
C                                                                       MCSP  520
      DO 65 K=1,NSAM                                                    MCSP  521
C                                                                       MCSP  522
      DO 65 L=2,N                                                       MCSP  523
      LM1 = L-1                                                         MCSP  524
C                                                                       MCSP  525
      DO 65 J=1,LM1                                                     MCSP  526
   65 VAR(L,J,K) = VAR(L,J,K)/ENS-ERR(L,K)*ERR(J,K)                     MCSP  527
C                                                                       MCSP  528
```

```
   66 CONTINUE                                                                  MCSP  529
C                                                                               MCSP  530
C                                                                               MCSP  531
      IF (IPRT.NE.0) GO TO 72                                                   MCSP  532
      WRITE (6,172)                                                             MCSP  533
C     WRITE GAINS, THEORETICAL COVARIANCES OF ESTIMATION ERROR                  MCSP  534
C     IF ONE SET OF GAINS HAS BEEN USED.                                        MCSP  535
      IF (IG.NE.0) GO TO 69                                                     MCSP  536
      WRITE (6,173)                                                             MCSP  537
C                                                                               MCSP  538
      DO 67 K=1,NSAM                                                            MCSP  539
      WRITE (6,174) K                                                           MCSP  540
C                                                                               MCSP  541
      DO 67 I=1,N                                                               MCSP  542
   67 WRITE (6,171) (GKS(I,J,K),J=1,M)                                          MCSP  543
C                                                                               MCSP  544
      WRITE (6,175)                                                             MCSP  545
C                                                                               MCSP  546
      DO 68 K=1,NSAM                                                            MCSP  547
      WRITE (6,176) K                                                           MCSP  548
C                                                                               MCSP  549
      DO 68 I=1,N                                                               MCSP  550
   68 WRITE (6,171) (PKKS(I,J,K),J=1,N)                                         MCSP  551
C                                                                               MCSP  552
   69 WRITE (6,181)                                                             MCSP  553
      WRITE (6,177)                                                             MCSP  554
      WRITE (6,178)                                                             MCSP  555
C                                                                               MCSP  556
      DO 70 K=1,NSAM                                                            MCSP  557
      WRITE (6,180)                                                             MCSP  558
C                                                                               MCSP  559
      DO 70 I=1,N                                                               MCSP  560
   70 WRITE (6,179) K,I,XM(I,K),ERR(I,K),VAR(I,I,K)                             MCSP  561
C                                                                               MCSP  562
      WRITE (6,181)                                                             MCSP  563
      IF (ISTAT.EQ.0) GO TO 72                                                  MCSP  564
      WRITE (6,182)                                                             MCSP  565
C                                                                               MCSP  566
      DO 71 K=1,NSAM                                                            MCSP  567
      WRITE (6,183) K                                                           MCSP  568
C                                                                               MCSP  569
      DO 71 I=1,N                                                               MCSP  570
   71 WRITE (6,171) (VAR(I,L,K),L=1,I)                                          MCSP  571
C                                                                               MCSP  572
      WRITE (6,181)                                                             MCSP  573
   72 IF (IPLT.NE.0) GO TO 88                                                   MCSP  574
C                                                                               MCSP  575
      DO 73 K=1,NSAM                                                            MCSP  576
```

```
   73 XP(K) = K                                                          MCSP  577
C                                                                        MCSP  578
      IF (IGPLT.NE.1) GO TO 76                                           MCSP  579
C                                                                        MCSP  580
      DO 75 I=1,N                                                        MCSP  581
C                                                                        MCSP  582
      DO 75 J=1,M                                                        MCSP  583
C                                                                        MCSP  584
      DO 74 K=1,NSAM                                                     MCSP  585
   74 YP(K) = GKS(I,J,K)                                                 MCSP  586
C                                                                        MCSP  587
      WRITE (6,181)                                                      MCSP  588
      CALL PLCTP (XP,YP,NSAM,0)                                          MCSP  589
   75 WRITE (6,184) I,J                                                  MCSP  590
C                                                                        MCSP  591
   76 IF (ITHVPL.NE.1) GO TO 79                                          MCSP  592
C                                                                        MCSP  593
      DO 78 I=1,N                                                        MCSP  594
C                                                                        MCSP  595
      DO 77 K=1,NSAM                                                     MCSP  596
   77 YP(K) = PKKS(I,I,K)                                                MCSP  597
C                                                                        MCSP  598
      WRITE (6,181)                                                      MCSP  599
      CALL PLCTP (XP,YP,NSAM,0)                                          MCSP  600
   78 WRITE (6,185) I,I                                                  MCSP  601
C                                                                        MCSP  602
   79 IF (IMTPLT.NE.1) GO TO 82                                          MCSP  603
C                                                                        MCSP  604
      DO 81 I=1,N                                                        MCSP  605
C                                                                        MCSP  606
      DO 80 K=1,NSAM                                                     MCSP  607
   80 YP(K) = XM(I,K)                                                    MCSP  608
C                                                                        MCSP  609
      WRITE (6,181)                                                      MCSP  610
      CALL PLCTP (XP,YP,NSAM,0)                                          MCSP  611
   81 WRITE (6,186) I                                                    MCSP  612
C                                                                        MCSP  613
   82 IF (ISMPLT.NE.1) GO TO 85                                          MCSP  614
C                                                                        MCSP  615
      DO 84 I=1,N                                                        MCSP  616
C                                                                        MCSP  617
      DO 83 K=1,NSAM                                                     MCSP  618
   83 YP(K) = ERR(I,K)                                                   MCSP  619
C                                                                        MCSP  620
      WRITE (6,181)                                                      MCSP  621
      CALL PLCTP (XP,YP,NSAM,0)                                          MCSP  622
   84 WRITE (6,187) I,I                                                  MCSP  623
C                                                                        MCSP  624
```

```
   85 IF (ISVPLT.NE.1) GO TO 88                                                    MCSP  625
C                                                                                 MCSP  626
      DO 87 I=1,N                                                                 MCSP  627
C                                                                                 MCSP  628
      DO 86 K=1,NSAM                                                              MCSP  629
   86 YP(K) = VAR(I,I,K)                                                          MCSP  630
C                                                                                 MCSP  631
      WRITE (6,181)                                                               MCSP  632
      CALL PLOTP (XP,YP,NSAM,0)                                                   MCSP  633
   87 WRITE (6,188) I                                                             MCSP  634
C                                                                                 MCSP  635
      WRITE (6,181)                                                               MCSP  636
C                                                                                 MCSP  637
C     THE FOLLOWING SECTION PLOTS THE DESIRED MISS DISTANCE INFORMATION.          MCSP  638
C                                                                                 MCSP  639
   88 IF (IMISS.NE.0) GO TO 105                                                   MCSP  640
C                                                                                 MCSP  641
      DO 90 K=1,NSAM                                                              MCSP  642
C                                                                                 MCSP  643
      DO 89 J=1,NENS                                                              MCSP  644
   89 SMISS(J,K) = MISSD(J,K)                                                     MCSP  645
C                                                                                 MCSP  646
   90 CONTINUE                                                                    MCSP  647
C                                                                                 MCSP  648
C                                                                                 MCSP  649
      DO 92 K=1,NSAM                                                              MCSP  650
C                                                                                 MCSP  651
      DO 91 J=1,NENS                                                              MCSP  652
   91 YP(K) = YP(K)+SMISS(J,K)                                                    MCSP  653
C                                                                                 MCSP  654
   92 CONTINUE                                                                    MCSP  655
C                                                                                 MCSP  656
C                                                                                 MCSP  657
      DO 93 K=1,NSAM                                                              MCSP  658
   93 YP(K) = YP(K)/ENS                                                           MCSP  659
C                                                                                 MCSP  660
      ENS1 = ENS-1.                                                               MCSP  661
C                                                                                 MCSP  662
      DO 95 K=1,NSAM                                                              MCSP  663
C                                                                                 MCSP  664
C                                                                                 MCSP  665
      DO 94 I=1,NENS                                                              MCSP  666
   94 TEMPA(K) = TEMPA(K)+(SMISS(I,K)-YP(K))**2                                   MCSP  667
C                                                                                 MCSP  668
   95 TEMPA(K) = SQRT(TEMPA(K)/ENS1)                                              MCSP  669
C                                                                                 MCSP  670
      IF (IMISSK.NE.0) GO TO 97                                                   MCSP  671
C                                                                                 MCSP  672
```

```
      DO 96 K=1,NSAM                                                    MCSP 673
   96 XP(K) = K                                                         MCSP 674
C                                                                       MCSP 675
      WRITE (6,181)                                                     MCSP 676
      CALL PLOTP (XP,YP,NSAM,0)                                         MCSP 677
      WRITE (6,199)                                                     MCSP 678
      WRITE (6,181)                                                     MCSP 679
      CALL PLOTP (XP,TEMPA,NSAM,0)                                      MCSP 680
      WRITE (6,200)                                                     MCSP 681
   97 IF (IMISSR.NE.0) GO TO 99                                         MCSP 682
C                                                                       MCSP 683
      DO 98 K=1,NSAM                                                    MCSP 684
   98 XP(K) = XS(1,K)                                                   MCSP 685
C                                                                       MCSP 686
      WRITE (6,181)                                                     MCSP 687
      CALL PLOTP (XP,YP,NSAM,0)                                         MCSP 688
      WRITE (6,201)                                                     MCSP 689
      WRITE (6,181)                                                     MCSP 690
      CALL PLOTP (XP,TEMPA,NSAM,0)                                      MCSP 691
      WRITE (6,202)                                                     MCSP 692
   99 IF (IMISSI.NE.0) GO TO 101                                        MCSP 693
C                                                                       MCSP 694
      DO 100 K=1,NSAM                                                   MCSP 695
  100 XINT(K) = XINT(K)/ENS                                             MCSP 696
C                                                                       MCSP 697
      WRITE (6,181)                                                     MCSP 698
      CALL PLOTP (XINT,YP,NSAM,0)                                       MCSP 699
      WRITE (6,203)                                                     MCSP 700
      WRITE (6,181)                                                     MCSP 701
      CALL PLOTP (XINT,TEMPA,NSAM,0)                                    MCSP 702
      WRITE (6,204)                                                     MCSP 703
      WRITE (6,181)                                                     MCSP 704
C                                                                       MCSP 705
C     THE FOLLOWING SECTION PRINTS THE DESIRED FIRING AND MISS DISTANCE MCSP 706
C     DATA.                                                             MCSP 707
C                                                                       MCSP 708
  101 IF (IFIRE.NE.0) GO TO 102                                         MCSP 709
      IF (NFIRE.EQ.0) GO TO 104                                         MCSP 710
      RDS = NFIRE/ENS                                                   MCSP 711
      IRDS = RDS+0.5                                                    MCSP 712
      HIT = NHIT/ENS                                                    MCSP 713
      IHIT = HIT+0.5                                                    MCSP 714
      SCARE = NSCARE/ENS                                                MCSP 715
      ISCARE = SCARE+0.5                                                MCSP 716
      AMISS = NMISS/ENS                                                 MCSP 717
      IMISSN = AMISS+0.5                                                MCSP 718
      PERHIT = (HIT/RDS)*100.                                           MCSP 719
      PERSCR = (SCARE/RDS)*100.                                         MCSP 720
```

```
      PERMIS = (AMISS/RDS)*100.                                          MCSP  721
      WRITE (6,195) IRDS                                                 MCSP  722
      WRITE (6,196) IHIT,PERHIT                                          MCSP  723
      WRITE (6,197) ISCARE,PERSCR                                        MCSP  724
      WRITE (6,198) IMISSN,PERMIS                                        MCSP  725
  102 IF (IMISSC.NE.0) GO TO 105                                         MCSP  726
      WRITE (6,181)                                                      MCSP  727
      WRITE (6,205)                                                      MCSP  728
C                                                                        MCSP  729
      DO 103 K=1,NSAM                                                    MCSP  730
  103 WRITE (6,206) K,YP(K),TEMPA(K)                                     MCSP  731
C                                                                        MCSP  732
      GO TO 105                                                          MCSP  733
  104 WRITE (6,207)                                                      MCSP  734
  105 CONTINUE                                                           MCSP  735
      STOP                                                               MCSP  736
C                                                                        MCSP  737
C                                                                        MCSP  738
C                                                                        MCSP  739
  106 FORMAT (5(I10))                                                    MCSP  740
  107 FORMAT (I2)                                                        MCSP  741
  108 FORMAT (7(I10))                                                    MCSP  742
  109 FORMAT (2(I5))                                                     MCSP  743
  110 FORMAT (5(I10))                                                    MCSP  744
  111 FORMAT (15I5)                                                      MCSP  745
  112 FORMAT (20X,'DESCRIPTION OF RUN',//)                               MCSP  746
  113 FORMAT (10X,'GAINS COMPUTED OFF-LINE AND READ IN',/)               MCSP  747
  114 FORMAT (10X,'GAINS COMPUTED ONCE IN "GAIN" BEFORE STARTING MONTE CMCSP  748
     1ARLO',/)                                                           MCSP  749
  115 FORMAT (10X,'GAINS COMPUTED FOR EACH MEMBER OF ENSEMBLE',/)        MCSP  750
  116 FORMAT (10X,'THE STANDARD LINEAR EQS. DO NOT CHARACTERIZE THE FILTMCSP  751
     1ER',/)                                                             MCSP  752
  117 FORMAT (10X,'THE STD. KALMAN EQS. CHARACTERIZE THE LINEAR FILTER',MCSP  753
     1/)                                                                 MCSP  754
  118 FORMAT (10X,'ONLY ONE TRACK IS USED AND IT IS GENERATED BY SUBROUTMCSP  755
     1INE TRACK',/)                                                      MCSP  756
  119 FORMAT (10X,'ONLY ONE TRACK IS USED AND IT IS READ IN',/)          MCSP  757
  120 FORMAT (10X,'SEVERAL TRACKS USED BUT NOT GENERATED FROM STD. LINEAMCSP  758
     1R DIFFERENCE EQS.',/)                                              MCSP  759
  121 FORMAT (10X,'SEVERAL TRACKS GENERATED BY USING THE STD. LINEAR DIFMCSP  760
     1FERENCE EQS.',/)                                                   MCSP  761
  122 FORMAT (10X,'MEAN OF TRACK, MEAN OF EST. ERROR AND COVARIANCE OF EMCSP  762
     1ST. ERROR ARE COMPUTED',/)                                         MCSP  763
  123 FORMAT (10X,'MEAN OF TRACK, MEAN AND VARRIANCES OF EST. ERROR ARE MCSP  764
     1COMPUTED',/)                                                       MCSP  765
  124 FORMAT (10X,'THE Q MATRIX IS COMPUTED ON-LINE AT EACH SAMPLE BY "QMCSP  766
     1ON"',/)                                                            MCSP  767
  125 FORMAT (10X,'THE COVARIANCE OF W IS READ IN AND Q IS COMPUTED BY "MCSP  768
```

```
   1CMAT" BEFORE STARTING MONTE CARLO',/)                                    MCSP  769
126 FORMAT (10X,'THE Q MATRIX IS READ IN',/)                                 MCSP  770
127 FORMAT (10X,'R IS COMPUTED ON-LINE AT EACH SAMPLE BY "RON"',/)           MCSP  771
128 FORMAT (10X,'R IS READ IN',/)                                            MCSP  772
129 FORMAT (////,20X,'INPUT DATA CALLED FOR',//)                             MCSP  773
130 FORMAT (10X,'PHI MATRIX',/)                                              MCSP  774
131 FORMAT (10X,'H MATRIX',/)                                                MCSP  775
132 FORMAT (10X,'R MATRIX',/)                                                MCSP  776
133 FORMAT (10X,'COVARIANCE OF W',/)                                         MCSP  777
134 FORMAT (10X,'Q MATRIX',/)                                                MCSP  778
135 FORMAT (10X,'GAMMA MATRIX',/)                                            MCSP  779
136 FORMAT (10X,'STANDARD DEVIATIONS OF MEASUREMENT NOISE',/)                MCSP  780
137 FORMAT (10X,'STANDARD DEVIATIONS OF INPUT FORCING W',/)                  MCSP  781
138 FORMAT (10X,'XHAT(0/-1)',/)                                              MCSP  782
139 FORMAT (10X,'MEANS AND VARIANCES OF X(0)',/)                             MCSP  783
140 FORMAT (10X,'P(0/-1)',/)                                                 MCSP  784
141 FORMAT (////,20X,'OUTPUT CALLED FOR',//)                                 MCSP  785
142 FORMAT (10X,'PRINTED OUTPUT OF THE FOLLOWING DATA',/)                    MCSP  786
143 FORMAT (15X,'GAIN MATRICES AND THEORETICAL COVARIANCE OF EST. ERROMCSP  787
   1R MATRICES',/)                                                           MCSP  788
144 FORMAT (15X,'SAMPLE MEANS OF TRACK AND ESTIMATION ERROR,SAMPLE VARMCSP  789
   1IANCES OF EST. ERROR',/)                                                 MCSP  790
145 FORMAT (15X,'COVARIANCE OF ESTIMATION ERROR MATRIX',/)                   MCSP  791
146 FORMAT (10X,'NO PRINTED MONTE CARLO OUTPUT IS DESIRED',//)               MCSP  792
147 FORMAT (10X,'THE FOLLOWING PLOTS ARE CALLED FOR',/)                      MCSP  793
148 FORMAT (15X,'G(K) VS. K',/)                                              MCSP  794
149 FORMAT (15X,'P(K/K) THEORETICAL VS. K',/)                                MCSP  795
150 FORMAT (15X,'MEAN OF TRACK VS. K',/)                                     MCSP  796
151 FORMAT (15X,'SAMPLE MEANS OF ESTIMATION ERROR VS. K',/)                  MCSP  797
152 FORMAT (15X,'SAMPLE VARIANCES OF ESTIMATION ERROR VS. K',/)              MCSP  798
153 FORMAT (10X,'NO PLOTTED MONTE CARLO OUTPUT IS CALLED FOR',/)             MCSP  799
154 FORMAT (20X,'INPUT DATA',///)                                            MCSP  800
155 FORMAT (4X,'N=',I2,4X,'M=',I2,4X'IN=',I2,4X'NSAM=',I3,4X,'NENS=',IMCSP  801
   15,4X'ND=',I2,//)                                                         MCSP  802
156 FORMAT (//,10X,'THE PHI MATRIX IS',/)                                    MCSP  803
157 FORMAT (//,10X,'THE H MATRIX IS',/)                                      MCSP  804
158 FORMAT (//,10X,'THE R MATRIX IS',/)                                      MCSP  805
159 FORMAT (//,10X,'THE COVARIANCE OF W MATRIX IS',/)                        MCSP  806
160 FORMAT (//,10X,'THE Q MATRIX IS',/)                                      MCSP  807
161 FORMAT (//,10X,'THE GAMMA MATRIX IS',/)                                  MCSP  808
162 FORMAT (//,10X,'THE MATRIX P(0/-1) IS',/)                                MCSP  809
163 FORMAT (//,10X,'THE STD. DEVIATIONS OF MEASUREMENT NOISE ARE',/)         MCSP  810
164 FORMAT (//,10X,'THE STD. DEVIATIONS OF INPUT FORCING W ARE',/)           MCSP  811
165 FORMAT (//,10X,'THE VECTOR XHAT(0/-1) IS',/)                             MCSP  812
166 FORMAT (//,10X,'THE MEAN OF THE VECTOR X(0) IS',/)                       MCSP  813
167 FORMAT (//,10X,'THE STANDARD DEVIATIONS OF THE VECTOR X(0) ARE',/)MCSP  814
168 FORMAT (//,10X,'THE INITIAL STATE IS',/)                                 MCSP  815
169 FORMAT (4(1PE20.12))                                                     MCSP  816
```

```
170 FORMAT (//,10X,'THE FIRST AND LAST POINTS ON THE SINGLE TRACK TO BMCSP 817
   1E USED ARE',/)                                                    MCSP 818
171 FORMAT (9(2X,1PE12.5),/)                                          MCSP 819
172 FORMAT ('1',20X,'OUTPUT DATA',///)                                MCSP 820
173 FORMAT (10X,'THE GAIN MATRICES ARE',/)                            MCSP 821
174 FORMAT (5X,'K=',I3,/,10X,'G(K)=',/)                               MCSP 822
175 FORMAT (1X,//,10X,'THE THEORETICAL COVARIANCE MATRIX IS',/)       MCSP 823
176 FORMAT (5X,'K=',I3,/,10X,'P(K/K)=',/)                            MCSP 824
177 FORMAT (T5,'TIME',T16,'VECTOR COM-',T34,'SAMPLE MEAN',            MCSP 825
   1 T51,'SAMPLE MEAN OF',T71,'SAMPLE VARIANCE OF')                   MCSP 826
178 FORMAT (T5,'INDEX',T16,'PONENT INDEX',T34,'OF TRACK',             MCSP 827
   1 T51,'ESTIMATION ERROR',T71,'ESTIMATION ERROR')                   MCSP 828
179 FORMAT (6X,I3,13X,I1,10X,1PE14.7,2(6X,1PE14.7))                   MCSP 829
180 FORMAT (//)                                                       MCSP 830
181 FORMAT ('1')                                                      MCSP 831
182 FORMAT (10X,'THE SAMPLE COVARIANCE OF EST. ERROR MATRIX IS',//)   MCSP 832
183 FORMAT (//,2X,'K=',I3,/)                                          MCSP 833
184 FORMAT (12X,'G(',I1,',',I1,')  VS. K')                            MCSP 834
185 FORMAT (12X,'PKK(',I1,',',I1,')  VS. K')                          MCSP 835
186 FORMAT (12X,'MEAN OF X(',I1,')  VS. K')                           MCSP 836
187 FORMAT (12X,'XHATKK(',I1,') -X(',I1,')  VS. K')                   MCSP 837
188 FORMAT (12X,'ERROR VARIANCE(',I1,') VS.. K')                      MCSP 838
189 FORMAT (15X,'MEAN AND STANDARD DEVIATION OF MISS DISTANCE VS. K',/MCSP 839
   1)                                                                 MCSP 840
190 FORMAT (15X,'MEAN AND STANDARD DEVIATION OF MISS DISTANCE VS. SLANMCSP 841
   1T RANGE',/)                                                       MCSP 842
191 FORMAT (15X,'MEAN AND STANDARD DEVIATION OF MISS DISTANCE VS AVERAMCSP 843
   1GE LENGTH OF PREDICTION INTERVAL',/)                              MCSP 844
192 FORMAT (15X,'AVERAGE NUMBER OF ROUNDS FIRED, HITS,SCARES,AND MISSEMCSP 845
   1S PER ENSEMBLE RUN',/)                                            MCSP 846
193 FORMAT (15X,'MEAN AND STANDARD DEVIATION OF THE MISS DISTANCE FOR MCSP 847
   1EACH K',/)                                                        MCSP 848
194 FORMAT (10X,'NO FIRING OR MISS DISTANCE DATA IS DESIRED',/)       MCSP 849
195 FORMAT (15X,'AVERAGE NUMBER OF ROUNDS FIRED PER ENSEMBLE RUN =',I3MCSP 850
   1,/)                                                               MCSP 851
196 FORMAT (15X,'AVERAGE NUMBER OF HITS PER ENSEMBLE RUN =',I3,2X,'OR'MCSP 852
   1,2X,F6.2,1X,'PERCENT',/)                                          MCSP 853
197 FORMAT (15X,'AVERAGE NUMBER OF SCARES PER ENSEMBLE RUN =',I3,2X,'CMCSP 854
   1R',F6.2,1X,'PERCENT',/)                                           MCSP 855
198 FORMAT (15X,'AVERAGE NUMBER OF MAGGIES DRAWERS PER ENSEMBLE RUN ='MCSP 856
   1,I3,2X,'OR',F6.2,1X,'PERCENT',/)                                  MCSP 857
199 FORMAT (15X,'MEAN OF MISS DISTANCE VS. K',/)                      MCSP 858
200 FORMAT (15X,'STANDARD DEVIATION OF MISS DISTANCE VS. K',/)        MCSP 859
201 FORMAT (15X,'MEAN OF MISS DISTANCE VS. SLANT RANGE',/)            MCSP 860
202 FORMAT (15X,'STANDARD DEVIATION OF MISS DISTANCE VS. SLANT RANGE',MCSP 861
   1/)                                                                MCSP 862
203 FORMAT (15X,'MEAN OF MISS DISTANCE VS. AVERAGE LENGTH OF PREDICTIOMCSP 863
   1N INTERVAL',/)                                                    MCSP 864
```

```
204 FORMAT (15X,'STANDARD DEVIATION OF MISS DISTANCE VS. AVERAGE LENGTMCSP  865
   1H OF PREDICTION INTERVAL',/)                                       MCSP  866
205 FORMAT (14X,'K',5X,'MEAN OF MISS DIST.',2X,'STD. DEV. OF MISS DISTMCSP  867
   1.',///)                                                            MCSP  868
206 FORMAT (12X,I3,7X,1PE13.6,10X,1PE13.6,/)                           MCSP  869
207 FORMAT (///,15X,'NO ROUNDS WERE FIRED; GO DIRECTLY TO THE SHIPYARDMCSP  870
   1, DO NOT COLLECT KILL',/)                                          MCSP  871
208 FORMAT (10X,'THE FOLLOWING MISS DISTANCE DATA IS PLOTTED',/)       MCSP  872
209 FORMAT (10X,'NO PLOTTED MISS DISTANCE DATA IS CALLED FOR',/)       MCSP  873
    END                                                                MCSP  874
```

```
      SUBROUTINE ESTIM                                                  ESTI    1
C     THIS SUBROUTINE UPDATES THE STATE ESTIMATE. IN THE DEFAULT        ESTI    2
C     CONDITION (IEST.EQ.0) THE STANDARD EQUATIONS                      ESTI    3
C                                                                       ESTI    4
C     XHAT(K/K)=XHAT(K/K-1)+G(K)*(Z(K)-H(5)*XHAT(K/K-1))                ESTI    5
C                                                                       ESTI    6
C     XHAT(K+1/K)=PHI*XHAT(K/K)                                         ESTI    7
C                                                                       ESTI    8
C     ARE EVALUATED                                                     ESTI    9
      REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI,PKKM10ESTI   10
     1,MISSD                                                            ESTI   11
      COMMON GAMMA(9,9),COVW(9,9),R(9,9),PHI(9,9),H(9,9),TEMP(9,9),TEMP1ESTI   12
     1(9,9),TEMP2(9,9),PKKM1(9,9),PKKM10(9,9),G(9,9),PKK(9,9),Q(9,9),EI(ESTI   13
     29,9),MISSD(100,160),VAR(9,9,160),GKS(9,9,160),PKKS(9,9,160),XM(9,1ESTI   14
     360),ERR(9,160),GAMMAS(9,9),PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGESTI   15
     4W(9),X(9),SIGXZ(9),XZMEAN(9),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SESTI   16
     5IGV(9),XHATZ(9),XINT(160),N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXESTI   17
     6Z,IV,IW,IEST,ND,NHIT,NSCARE,NMISS,NFIRE                           ESTI   18
C     TAKE THE APPROPRIATE GAIN AND STORE IN THE ARRAY GK               ESTI   19
C                                                                       ESTI   20
C                                                                       ESTI   21
      DO 1 I=1,N                                                        ESTI   22
C                                                                       ESTI   23
C                                                                       ESTI   24
      DO 1 J=1,M                                                        ESTI   25
    1 GK(I,J) = GKS(I,J,K)                                              ESTI   26
C                                                                       ESTI   27
C                                                                       ESTI   28
      IF (IEST.NE.0) GO TO 2                                            ESTI   29
      CALL VPROD (HS,XHKKM1,M,N,VTMP)                                   ESTI   30
      CALL VSUB (Z,VTMP,M,VTMP)                                         ESTI   31
      CALL VPROD (GK,VTMP,N,M,VTMP)                                     ESTI   32
      CALL VADD (XHKKM1,VTMP,N,XHKK)                                    ESTI   33
C                                                                       ESTI   35
C     XHAT(K/K) HAS BEEN COMPUTED AND STORED IN THE ARRAY XHKK          ESTI   36
      CALL VPROD (PHIS,XHKK,N,N,XHKKM1)                                 ESTI   37
C                                                                       ESTI   39
C     XHAT(K+1/K) HAS BEEN COMPUTED AND STORED IN THE ARRAY XHKKM1      ESTI   40
      RETURN                                                            ESTI   41
    2 CONTINUE                                                          ESTI   42
C     IF STANDARD EQUATIONS ARE NOT TO BE USED, THE APPROPRIATE         ESTI   43
C     EQUATIONS MUST BE INSERTED HERE BY THE USER.                      ESTI   44
      RETURN                                                            ESTI   45
      END                                                               ESTI   46
```

```
      SUBROUTINE GAIN                                                    GAIN    1
      REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI,PKKM10GAIN    2
     1,MISSD                                                             GAIN    3
      COMMON GAMMA(9,9),COVW(9,9),R(9,9),PHI(9,9),H(9,9),TEMP(9,9),TEMP1GAIN    4
     1(9,9),TEMP2(9,9),PKKM1(9,9),PKKM10(9,9),G(9,9),PKK(9,9),Q(9,9),EI(GAIN    5
     29,9),MISSD(100,160),VAR(9,9,160),GKS(9,9,160),PKKS(9,9,160),XM(9,1GAIN    6
     360),ERR(9,160),GAMMAS(9,9),PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGGAIN    7
     4W(9),X(9),SIGXZ(9),XZMEAN(9),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SGAIN    8
     5IGV(9),XHATZ(9),XINT(160),N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXGAIN    9
     6Z,IV,IW,IEST,ND,NHIT,NSCARE,NMISS,NFIRE                            GAIN   10
C                                                                        GAIN   11
C     G(K) = P(K/K-1)*HT*(H*P(K/K-1)*HT + R)                             GAIN   12
      IF (K.NE.1) GO TO 2                                                GAIN   13
C                                                                        GAIN   14
C                                                                        GAIN   15
      DO 1 IL=1,N                                                        GAIN   16
C                                                                        GAIN   17
      DO 1 JL=1,N                                                        GAIN   18
    1 PKKM1(IL,JL) = PKKM10(IL,JL)                                       GAIN   19
C                                                                        GAIN   20
C                                                                        GAIN   21
    2 CONTINUE                                                           GAIN   22
      CALL TRANS (H,M,N,TEMP2)                                           GAIN   23
      CALL PROD (PKKM1,TEMP2,N,N,M,TEMP)                                 GAIN   24
      CALL PROD (H,TEMP,M,N,M,TEMP1)                                     GAIN   25
      CALL ADD (TEMP1,R,M,M,TEMP1)                                       GAIN   26
      IF (M.EQ.1) GO TO 4                                                GAIN   27
      MD = ND                                                            GAIN   28
      CALL GAUSS3 (M,EPS,TEMP1,TEMP2,KER,MD)                             GAIN   29
      CALL PROD (TEMP,TEMP2,N,M,M,G)                                     GAIN   30
C                                                                        GAIN   31
C     NOTE HERE  PKK(I,J) = P(K/K)  WHERE                                GAIN   32
C     P(K/K) = (I-G(K)*H)*P(K/K-1)                                       GAIN   33
    3 CALL PROD (G,H,N,M,N,TEMP)                                         GAIN   34
      CALL SUB (EI,TEMP,N,N,TEMP2)                                       GAIN   35
      CALL PROD (TEMP2,PKKM1,N,N,N,PKK)                                  GAIN   36
C                                                                        GAIN   37
C     NOTE HERE  PKKM1(I,J) = P(K/K-1)  WHERE                            GAIN   38
C     P(K/K-1)= PHI*P(K-1/K-1)*PHIT + Q                                  GAIN   39
      CALL TRANS (PHI,N,N,TEMP2)                                         GAIN   40
      CALL PROD (PKK,TEMP2,N,N,N,TEMP)                                   GAIN   41
      CALL PROD (PHI,TEMP,N,N,N,TEMP1)                                   GAIN   42
      CALL ADD (TEMP1,Q,N,N,PKKM1)                                       GAIN   43
      RETURN                                                             GAIN   44
C                                                                        GAIN   45
C                                                                        GAIN   46
    4 DO 5 I=1,N                                                         GAIN   47
    5 G(I,1) = TEMP(I,1)/TEMP1(1,1)                                      GAIN   48
```

```
C
C
      GO TO 3                                                          GAIN   49
      END                                                              GAIN   50
                                                                       GAIN   51
                                                                       GAIN   52
```

```
      SUBROUTINE QMAT                                                   QMAT    1
C                                                                       QMAT    2
CC    THIS SUBROUTINE COMPUTES THE MATRIX Q FROM THE EQUATION           QMAT    3
CCC                                                                     QMAT    4
CCC      Q=GAMMA* E(W*WT) * GAMMAT                                      QMAT    5
CCCC                                                                    QMAT    6
CCCCC DOUBLE PRECISION ARITHMETIC IS USED                               QMAT    7
CCCC                                                                    QMAT    8
CC                                                                      QMAT    9
      REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI,PKKM10QMAT   10
     1,MISSD                                                            QMAT   11
      COMMON GAMMA(9,9),COVW(9,9),R(9,9),PHI(9,9),H(9,9),TEMP(9,9),TEMP1QMAT   12
     1(9,9),TEMP2(9,9),PKKM1(9,9),PKKM10(9,9),G(9,9),PKK(9,9),Q(9,9),EI(QMAT   13
     29,9),MISSD(100,160),VAR(9,9,160),GKS(9,9,160),PKKS(9,9,160),XM(9,1QMAT   14
     360),ERR(9,160),GAMMAS(9,9),PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGQMAT   15
     4W(9),X(9),SIGXZ(9),XZMEAN(9),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SQMAT   16
     5IGV(9),XHATZ(9),XINT(160),N,NSAM,IC,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXCMAT   17
     6Z,IV,IW,IEST,ND,NHIT,NSCARE,NMISS,NFIRE                           QMAT   18
C                                                                       QMAT   19
CC                                                                      QMAT   20
      CALL PROD (GAMMA,COVW,N,IN,IN,TEMP)                               QMAT   21
      CALL TRANS (GAMMA,N,IN,TEMP1)                                     QMAT   22
      CALL PROD (TEMP,TEMP1,N,IN,N,G)                                   QMAT   23
      RETURN                                                            QMAT   24
      END                                                               QMAT   25
```

```
      SUBROUTINE MEAS                                                                    MEAS    1
      REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI,PKKM10MEAS    2
     1,MISSD                                                                             MEAS    3
      COMMON GAMMA(9,9),COVW(9,9),R(9,9),PHI(9,9),H(9,9),TEMP(9,9),TEMP1MEAS    4
     1(9,9),TEMP2(9,9),PKKM1(9,9),PKKM10(9,9),G(9,9),PKK(9,9),Q(9,9),EI(MEAS    5
     29,9),MISSD(100,160),VAR(9,9,160),GKS(9,9,160),PKKS(9,9,160),XM(9,1MEAS    6
     360),ERR(9,160),GAMMAS(9,9),PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGMEAS    7
     4W(9),X(9),SIGXZ(9),XZMEAN(9),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SMEAS    8
     51GV(9),XHATZ(9),XINT(160),N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRC,IXMEAS    9
     6Z,IV,IW,IEST,ND,NHIT,NSCARE,NMISS,NFIRE                                          MEAS   10
      N1 = N/3                                                                         MEAS   11
      NN = N-1                                                                         MEAS   12
      J = 0                                                                            MEAS   13
C                                                                                      MEAS   14
      DO 1 I=1,NN,N1                                                                   MEAS   15
      J = J+1                                                                          MEAS   16
    1 Z(J) = XS(I,K)                                                                   MEAS   17
C                                                                                      MEAS   18
      CALL SNORM (IV,V,M)                                                              MEAS   19
C                                                                                      MEAS   20
C                                                                                      MEAS   21
      DO 2 I=1,M                                                                       MEAS   22
    2 V(I) = SIGV(I)*V(I)                                                              MEAS   23
C                                                                                      MEAS   24
C                                                                                      MEAS   25
      CALL VADD (Z,V,M,Z)                                                              MEAS   26
      RETURN                                                                           MEAS   28
      END                                                                              MEAS   29
```

```
      SUBROUTINE KILL                                                   KILL    1
      REAL *8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI,PKKM1OKILL    2
     1,MISSD,TEMPP1,PREPOS,VDTEMP,ER1,DELTAT,DELTAP,DELTR,D,CHORD1,ANGLEKILL    3
     2A,CHORD2,ANGLE8,ELEG1,ERRNET,PI,DEGRAD,B,TEMPX,TEMPX1,SUM1,SUM2   KILL    4
      COMMON GAMMA(9,9),COVW(9,9),R(9,9),PHI(9,9),H(9,9),TEMP(9,9),TEMP1KILL    5
     1(9,9),TEMP2(9,9),PKKM1(9,9),PKKM10(9,9),G(9,9),PKK(9,9),Q(9,9),EI(KILL    6
     29,9),MISSD(100,160),VAR(9,9,160),GKS(9,9,160),PKKS(9,9,160),XM(9,1KILL    7
     360),ERR(9,160),GAMMAS(9,9),PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGKILL    8
     4W(9),X(9),SIGXZ(9),XZMEAN(9),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SKILL    9
     5IGV(9),XHATZ(9),XINT(160),N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXKILL   10
     6Z,IV,IW,IEST,ND,NHIT,NSCARE,NMISS,NFIRE                           KILL   11
      DIMENSION TEMPP1(9,9),TEMPX(9),PREPOS(9,160),VDTEMP(9),ER1(3),TEMP
     1X1(9)
      DATA PI,DEGRAD,SAMPR,RING1,RING2/3.14159292635,57.2957795131,0.25,
     18.,16./
      N1 = N/3                                                          KILL   17
      N2 = 2*N1                                                         KILL   18
      N3 = N1+1                                                         KILL   19
      NN = N-1                                                          KILL   20
C                                                                       KILL   21
CC    STORE THE CURRENT ESTIMATE XHAT(K/K) AND THE EXACT POSITION X(K)  KILL   22
CC    FOR FUTURE USE.  CONVERT ALL ANGLES TO RADIAN MEASURE.            KILL   23
CC                                                                      KILL   24
C                                                                       KILL   25
      DO 1 I=1,N                                                        KILL   26
      VDTEMP(I) = 0.0D0                                                 KILL   27
      TEMPX1(I) = 0.0D0                                                 KILL   28
      TEMPX1(I) = XS(I,K)                                               KILL   29
    1 PREPOS(I,K) = XHKK(I)                                             KILL   30
C                                                                       KILL   31
C                                                                       KILL   32
      DO 2 I=N3,N                                                       KILL   33
      TEMPX1(I) = TEMPX1(I)/DEGRAD                                      KILL   34
    2 PREPOS(I,K) = PREPOS(I,K)/DEGRAD                                  KILL   35
C                                                                       KILL   36
CC                                                                      KILL   37
CC    THIS SECTION COMPUTES THE TIME OF FLIGHT (TOF) OF THE PROJECTILE, KILL   38
C     FROM DATA GIVEN IN THE 5"/54 CAL RANGE TABLE.                     KILL   39
      TOF = (.143E-02)*XHKK(1)
C     DURING THE PROJECTILE TOF, N MEASUREMENTS WILL BE TAKEN.          KILL   49
      NSAMP = TOF/SAMPR
      TNSAMP = FLOAT(NSAMP)
C                                                                       KILL   48
CC                                                                      KILL   53
CC    IF THE PROJECTILE TOF IS GREATER THAN 10.00 SECS., THE TARGET     KILL   54
CC    WILL BE CONSIDERED TO BE OUT OF RANGE.  NO ROUNDS WILL BE FIRED.  KILL   55
CC    IF THE TARGET IS NOT TAKEN UNDER FIRE, ASSIGN A LARGE VALUE OF    KILL   56
C     MISS DISTANCE AS A PENALTY.                                       KILL   57
```

```
C
      IF(NSAMP.GE.K) GO TO 3                                            KILL  58
      IF(TOF.LE.10.0) GO TO 31                                          KILL  59
    3 ERRNET = 6.0D 02                                                  KILL  60
      GO TO 17                                                          KILL  61
   31 CONTINUE                                                          KILL  62
      XINT(K) = XINT(K)+TNSAMP                                          KILL  63
C                                                                       KILL  64
C                                                                       KILL  65
      DO 4 I=1,N                                                        KILL  66
    4 TEMPX(I) = PREPOS(I,K-NSAMP)                                      KILL  67
C                                                                       KILL  68
CC                                                                      KILL  69
CC    THIS SECTION RAISES PHI TO THE NSAMP POWER.                       KILL  70
CC                                                                      KILL  71
C                                                                       KILL  72
      DO 5 I=1,N                                                        KILL  73
C                                                                       KILL  74
      DO 5 J=1,N                                                        KILL  75
    5 TEMPP1(I,J) = 0.0D0                                               KILL  76
C                                                                       KILL  77
CC                                                                      KILL  78
      TEMPP1(1,1) = 1.0D0                                               KILL  79
      TEMPP1(1,2) = TNSAMP*SAMPR                                        KILL  80
      TEMPP1(2,2) = 1.0D0                                               KILL  81
      IF (N.NE.9) GO TO 8                                               KILL  82
      TEMPP1(3,3) = PHI(3,3)**NSAMP                                     KILL  83
      SUM1 = 0.00                                                       KILL  84
      SUM2 = 0.00                                                       KILL  85
      NSAMP1 = NSAMP-1                                                  KILL  86
C                                                                       KILL  87
CC                                                                      KILL  88
      DO 6 I=1,NSAMP                                                    KILL  89
      J = I-1                                                           KILL  90
    6 SUM1 = SUM1+(PHI(3,3)**J)                                         KILL  91
C                                                                       KILL  92
CC                                                                      KILL  93
CC                                                                      KILL  94
      DO 7 I=1,NSAMP1                                                   KILL  95
      J = I-1                                                           KILL  96
    7 SUM2 = SUM2+(TNSAMP-I)*(PHI(3,3)**J)                              KILL  97
C                                                                       KILL  98
CC                                                                      KILL  99
      TEMPP1(2,3) = PHI(2,3)*SUM1                                       KILL 100
      TEMPP1(1,3) = PHI(1,3)*SUM1+PHI(1,2)*PHI(2,3)*SUM2                KILL 101
C                                                                       KILL 102
CC                                                                      KILL 103
    8 DO 10 NA=N1,N2,N1                                                 KILL 104
                                                                        KILL 105
```

```
C                                                                       KILL 106
      DO 9 I=1,N1                                                       KILL 107
      INA = I+NA                                                        KILL 108
C                                                                       KILL 109
      DO 9 J=1,N1                                                       KILL 110
      JNA = J+NA                                                        KILL 111
    9 TEMPP1(INA,JNA) = TEMPP1(I,J)                                     KILL 112
C                                                                       KILL 113
   10 CONTINUE                                                          KILL 114
C                                                                       KILL 115
CC                                                                      KILL 116
CCC   FORM THE PREDICTED POSITION, XHAT(K/K-NSAMP) = (PHI**NSAMP)*      KILL 117
CCC   XHAT(K-NSAMP/K-NSAMP)                                             KILL 118
C                                                                       KILL 119
      DO 12 I=1,N                                                       KILL 120
C                                                                       KILL 121
      DO 11 J=1,N                                                       KILL 122
   11 VDTEMP(I) = VDTEMP(I)+TEMPP1(I,J)*TEMPX(J)                        KILL 123
C                                                                       KILL 124
   12 CONTINUE                                                          KILL 125
C                                                                       KILL 126
CC                                                                      KILL 127
CCC   COMPUTE THE PREDICTION ERRORS AND THE MAGNITUDE OF THE ERROR      KILL 128
CCC   VECTOR.                                                           KILL 129
CC                                                                      KILL 130
      J = 0                                                             KILL 131
C                                                                       KILL 132
      DO 13 I=1,NN,N1                                                   KILL 133
      J = J+1                                                           KILL 134
   13 ER1(J) = TEMPX1(I)-VDTEMP(I)                                      KILL 135
C                                                                       KILL 163
CC                                                                      KILL 136
CCC                                                                     KILL 137
      DELTR = DABS(ER1(1))                                              KILL 138
      DELTAP = DABS(ER1(2))                                             KILL 139
      DELTAT = DABS(ER1(3))                                             KILL 140
      D = DMAX1(TEMPX1(1),VDTEMP(1))                                    KILL 141
      CHORD1 = 2.*DSIN(DELTAP/2.)*D                                     KILL 142
      ANGLEA = (PI-DELTAP)/2.                                           KILL 143
      ELEG1 = ((CHORD1**2)+(DELTR**2)-2.*DELTR*CHORD1*DCOS(ANGLEA))     KILL 144
      ANGLEB = (PI-DELTAT)/2.                                           KILL 145
      CHORD2 = 2.*DSIN(DELTAT/2.)*D                                     KILL 146
      B = DSQRT(ELEG1)                                                  KILL 147
      ERRNET = DSQRT(ELEG1+((CHORD2)**2)-2.*B*(CHORD2)*DCOS(ANGLEB))    KILL 148
C                                                                       KILL 150
CC    COMPUTE ROUNDS FIRED, HITS, MISSES AND SCARES                     KILL 151
CC                                                                      KILL 152
      IF (ERRNET.GT.RING2) GO TO 15                                     KILL 153
```

```
      IF (ERRNET.LE.RING1) GO TO 14                                     KILL  154
      NSCARE = NSCARE+1                                                 KILL  155
      GO TO 16                                                          KILL  156
   14 NHIT = NHIT+1                                                     KILL  157
      GO TO 16                                                          KILL  158
   15 NMISS = NMISS+1                                                   KILL  159
   16 NFIRE = NFIRE+1                                                   KILL  160
   17 MISSD(ITER,K) = ERRNET                                            KILL  161
      RETURN                                                            KILL  162
      END                                                               KILL  166
```

```
      SUBROUTINE QON                                                    QONL    1
C                                                                       QONL    2
CCC   IF Q IS TO BE COMPUTED ON-LINE (IFLQ.NE.0) IT IS DONE             QONL    3
CCC   IN THIS SUBROUTINE                                                QONL    4
C                                                                       QONL    5
      REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI,PKKM10QONL    6
     1,MISSD                                                            QONL    7
      COMMON GAMMA(9,9),COVW(9,9),R(9,9),PHI(9,9),H(9,9),TEMP(9,9),TEMP1QONL    8
     1(9,9),TEMP2(9,9),PKKM1(9,9),PKKM10(9,9),G(9,9),PKK(9,9),Q(9,9),EI(QONL    9
     29,9),MISSD(100,160),VAR(9,9,160),GKS(9,9,160),PKKS(9,9,160),XM(9,1QONL   10
     360),ERR(9,160),GAMMAS(9,9),PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGQONL   11
     4W(9),X(9),SIGXZ(9),XZMEAN(9),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SQONL   12
     5IGV(9),XHATZ(9),XINT(160),N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXQONL   13
     6Z,IV,IW,IEST,ND,NHIT,NSCARE,NMISS,NFIRE                           QONL   14
C                                                                       QONL   15
CCC   THE APPROPRIATE STATEMENTS FOR COMPUTING Q ON-LINE MUST           QONL   16
CCC   BE INSERTED HERE BY THE USER                                      QONL   17
C                                                                       QONL   18
      RETURN                                                            QONL   19
      END                                                               QONL   20
```

```
      SUBROUTINE RON                                                    RONL    1
C                                                                       RONL    2
C     IF R IS TO BE COMPUTED ON-LINE (IFLR.NE.0) IT IS DONE             RONL    3
C     IN THIS SUBROUTINE                                                RONL    4
C                                                                       RONL    5
      REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI,PKKM10RONL    6
     1,MISSD                                                            RONL    7
      COMMON GAMMA(9,9),COVW(9,9),R(9,9),PHI(9,9),H(9,9),TEMP(9,9),TEMP1RONL    8
     1(9,9),TEMP2(9,9),PKKM1(9,9),PKKM10(9,9),G(9,9),PKK(9,9),Q(9,9),EI(RONL    9
     29,9),MISSD(100,160),VAR(9,9,160),GKS(9,9,160),PKKS(9,9,160),XM(9,1RONL   10
     360),ERR(9,160),GAMMAS(9,9),PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGRONL   11
     4W(9),X(9),SIGXZ(9),XZMEAN(9),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SRONL   12
     5IGV(9),XHATZ(9),XINT(160),N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRC,IXRONL   13
     6Z,IV,IW,IEST,ND,NHIT,NSCARE,NMISS,NFIRE                           RONL   14
C                                                                       RONL   15
C     THE APPROPRIATE STATEMENTS FOR COMPUTING R ON-LINE MUST           RONL   16
C     BE INSERTED HERE BY THE USER                                      RONL   17
C                                                                       RONL   18
      RETURN                                                            RONL   19
      END                                                               RONL   20
```

```
      SUBROUTINE STAT                                                   STAT    1
C     THIS SUBROUTINE  COMPUTES RUNNING SUMS USED IN DETERMINING THE    STAT    2
C     SAMPLE STATISTICS OF TRACK AND ESTIMATION ERRORS. IN THE DEFAULT   STAT    3
C     OPTION (ISTAT.EQ.0) THE STATISTICS TO BE COMPUTED ARE MEAN OF      STAT    4
C     TRACK, MEAN OF ESTIMATION ERROR AND VARIANCE OF ESTIMATION         STAT    5
C     ERROR. IF(ISTAT.NE.0) THE OFF-DIAGONAL TERMS IN THE COVARIANCE OF  STAT    6
C     ESTIMATION ERROR MATRIX ARE ALSO COMPUTED.                         STAT    7
      REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI,PKKM10STAT    8
     1,MISSD                                                            STAT    9
      COMMON GAMMA(9,9),COVW(9,9),R(9,9),PHI(9,9),H(9,9),TEMP(9,9),TEMP1STAT   10
     1(9,9),TEMP2(9,9),PKKM1(9,9),PKKM10(9,9),G(9,9),PKK(9,9),Q(9,9),EI(STAT   11
     29,9),MISSD(100,160),VAR(9,9,160),GKS(9,9,160),PKKS(9,9,160),XM(9,1STAT   12
     360),ERR(9,160),GAMMAS(9,9),PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGSTAT   13
     4W(9),X(9),SIGXZ(9),XZMEAN(9),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SSTAT   14
     5IGV(9),XHATZ(9),XINT(160),N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXSTAT   15
     6Z,IV,IW,IEST,ND,NHIT,NSCARE,NMISS,NFIRE                           STAT   16
      DIMENSION EXH(9)                                                  STAT   17
      IF (ITRK.NE.1) GO TO 2                                            STAT   18
      IF (ITER.NE.1) GO TO 4                                            STAT   19
C                                                                       STAT   20
C                                                                       STAT   21
      DO 1 J=1,N                                                        STAT   22
    1 XM(J,K) = XS(J,K)                                                 STAT   23
C                                                                       STAT   24
C                                                                       STAT   25
      GO TO 4                                                           STAT   26
    2 CONTINUE                                                          STAT   27
C                                                                       STAT   28
C                                                                       STAT   29
      DO 3 J=1,N                                                        STAT   30
    3 XM(J,K) = XM(J,K)+XS(J,K)                                         STAT   31
C                                                                       STAT   32
C                                                                       STAT   33
    4 CONTINUE                                                          STAT   34
C                                                                       STAT   35
C                                                                       STAT   36
      DO 5 J=1,N                                                        STAT   37
      EXH(J) = XHKK(J)-XS(J,K)                                          STAT   38
      ERR(J,K) = ERR(J,K)+EXH(J)                                        STAT   39
    5 VAR(J,J,K) = VAR(J,J,K)+EXH(J)**2                                 STAT   40
C                                                                       STAT   41
C                                                                       STAT   42
      IF (ISTAT.EQ.0) RETURN                                            STAT   43
C                                                                       STAT   44
C                                                                       STAT   45
      DO 6 L=2,N                                                        STAT   46
      LM1 = L-1                                                         STAT   47
C                                                                       STAT   48
```

```
C
      DO 6 J=1,LM1                                                      STAT   49
    6 VAR(L,J,K) = VAR(L,J,K)+EXH(L)*EXH(J)                             STAT   50
C                                                                       STAT   51
C                                                                       STAT   52
      RETURN                                                            STAT   53
      END                                                               STAT   54
                                                                        STAT   55
```

```
      SUBROUTINE TRACK                                                  TRCK    1
C     IF TRACK IS TO BE GENERATED ON-LINE IT IS DONE IN THIS SUBROUTINE TRCK    2
C     IN THE DEFAULT OPTION (ITRK.EQ.0) THE TRACK IS GENERATED          TRCK    3
C     FROM THE STANDARD LINEAR DIFFERENCE EQUATION                      TRCK    4
C                                                                       TRCK    5
C     X(K+1)=PHI*X(K)+GAMMA*W(K)                                        TRCK    6
C                                                                       TRCK    7
      REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI,PKKM1OTRCK    8
     1,MISSD                                                            TRCK    9
      COMMON GAMMA(9,9),COVW(9,9),R(9,9),PHI(9,9),H(9,9),TEMP(9,9),TEMP1TRCK   10
     1(9,9),TEMP2(9,9),PKKM1(9,9),PKKM10(9,9),G(9,9),PKK(9,9),Q(9,9),EI(TRCK   11
     29,9),MISSD(100,160),VAR(9,9,160),GKS(9,9,160),PKKS(9,9,160),XM(9,1TRCK   12
     360),ERR(9,160),GAMMAS(9,9),PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGTRCK   13
     4W(9),X(9),SIGXZ(9),XZMEAN(9),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),STRCK   14
     5IGV(9),XHATZ(9),XINT(100),N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXTRCK   15
     6Z,IV,IW,IEST,ND,NHIT,NSCARE,NMISS,NFIRE                           TRCK   16
      DIMENSION W(9)                                                    TRCK   17
C     ITRK NE.0 OR 1 -- SEVERAL TRACKS GENERATED,BUT NOT FROM STD.      TRCK   18
C                       LINEAR EQS.                                     TRCK   19
C         = 0 -- SEVERAL TRACKS GENERATED FROM STD LINEAR EQS           TRCK   20
C         = 1 -- ONLY ONE TRACK IS USED                                 TRCK   21
      IF (ITRK.NE.0) GO TO 4                                            TRCK   22
      CALL SNORM (IW,W,IN)                                              TRCK   23
C     CONVERT EACH N(0,1) R.V. TO N(0,SIGW(I)) R.V.                     TRCK   24
C                                                                       TRCK   25
C                                                                       TRCK   26
      DO 1 I=1,IN                                                       TRCK   27
    1 W(I) = SIGW(I)*W(I)                                               TRCK   28
C                                                                       TRCK   29
C                                                                       TRCK   30
C                                                                       TRCK   31
C                                                                       TRCK   32
      DO 2 I=1,N                                                        TRCK   33
    2 X(I) = XS(I,K)                                                    TRCK   34
C                                                                       TRCK   35
C                                                                       TRCK   36
      CALL VPROD (GAMMAS,W,N,IN,W)                                      TRCK   37
      CALL VPROD (PHIS,X,N,N,VTMP)                                      TRCK   38
      CALL VADD (VTMP,W,N,VTMP)                                         TRCK   39
C                                                                       TRCK   40
C                                                                       TRCK   41
      DO 3 I=1,N                                                        TRCK   42
    3 XS(I,K+1) = VTMP(I)                                               TRCK   43
C                                                                       TRCK   44
C                                                                       TRCK   45
C     NEW VALUE OF X HAS BEEN COMPUTED AND STORED IN THE ARRAY XS       TRCK   46
      RETURN                                                            TRCK   47
    4 IF (ITRK.NE.1) GO TO 6                                            TRCK   48
```

```
C       IF(ITRK.EQ.1) THE USER MUST INSERT HERE THE STATEMENTS REQUIRED   TRCK   49
C     TO GENERATE A SINGLE TRAJECTORY AND STORE IT IN THE ARRAY           TRCK   50
C     XS(I,K),I=1,N,K=2,NSAM (NOTE THAT IF A SINGLE TRAJECTORY IS TO BE   TRCK   51
C     GENERATED, THE INITIAL CONDITION HAS BEEN READ IN AND STORED        TRCK   52
C     IN XS(I,1),I=1,N)                                                   TRCK   53
C                                                                         TRCK   54
C                                                                         TRCK   55
      DO 5 K=2,NSAM                                                       TRCK   56
      EKM1 = K-1                                                          TRCK   57
      T = 0.5*EKM1                                                        TRCK   58
      XS(1,K) = 13.-0.0173205*T                                           TRCK   59
      XS(2,K) = -0.0173205                                                TRCK   60
      XS(3,K) = 7.7-0.01*T                                                TRCK   61
    5 XS(4,K) = -0.01                                                     TRCK   62
C                                                                         TRCK   63
C                                                                         TRCK   64
      RETURN                                                              TRCK   65
    6 CONTINUE                                                            TRCK   66
C     IF THIS POINT IS REACHED, ITRK NOT EQUAL 0 OR 1 INDICATING THAT     TRCK   67
C     SEVERAL TRACKS ARE TO BE GENERATED, BUT NOT BY USING THE STD.       TRCK   68
C     LINEAR DIFFERENCE EQS..THE USER MUST SUPPLY THE APPROPRIATE         TRCK   69
C     STATEMENTS HERE.                                                    TRCK   70
      RETURN                                                              TRCK   71
      END                                                                 TRCK   72
```

```
      SUBROUTINE XZERO                                                  XZER    1
C     THIS SUBROUTINE GENERATES THE INITIAL STATE VALUE FROM A NORMAL   XZER    2
C     RANDOM NUMBER GENERATOR. IT IS ASSUMED THAT THE INITIAL STATE     XZER    3
C     HAS COMPONENTS THAT ARE INDEPENDENT                               XZER    4
      REAL *8GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI,PKKM10XZER    5
     1,MISSD                                                            XZER    6
      COMMON GAMMA(9,9),COVW(9,9),R(9,9),PHI(9,9),H(9,9),TEMP(9,9),TEMP1XZER    7
     1(9,9),TEMP2(9,9),PKKM1(9,9),PKKM10(9,9),G(9,9),PKK(9,9),Q(9,9),EI(XZER    8
     29,9),MISSD(100,160),VAR(9,9,160),GKS(9,9,160),PKKS(9,9,160),XM(9,1XZER    9
     360),ERR(9,160),GAMMAS(9,9),PHIS(9,9),XS(9,160),HS(9,9),GK(9,9),SIGXZER   10
     4W(9),X(9),SIGXZ(9),XZMEAN(9),XHKK(9),XHKKM1(9),VTMP(9),Z(9),V(9),SXZER   11
     5IGV(9),XHATZ(9),XINT(160),N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXXZER   12
     6Z,IV,IW,IEST,ND,NHIT,NSCARE,NMISS,NFIRE                           XZER   13
      CALL SNORM (IXZ,X,N)                                              XZER   14
C                                                                       XZER   15
C                                                                       XZER   16
      DO 1 I=1,N                                                        XZER   17
    1 XS(I,1) = SIGXZ(I)*X(I)+XZMEAN(I)                                 XZER   18
C                                                                       XZER   19
C                                                                       XZER   20
      RETURN                                                            XZER   21
      END                                                               XZER   22
```

```
      SUBROUTINE ADD (A,B,N,M,C)                                        MADD    1
C     THIS SUBROUTINE ADDS THE NXM MATRICES A AND B, STORING THE        MADD    2
C     RESULT IN C                                                       MADD    3
      REAL *8A,B,C                                                      MADD    4
      DIMENSION A(9,9), B(9,9), C(9,9)                                  MADD    5
C                                                                       MADD    6
C                                                                       MADD    7
      DO 1 I=1,N                                                        MADD    8
C                                                                       MADD    9
C                                                                       MADD   10
      DO 1 J=1,M                                                        MADD   11
    1 C(I,J) = A(I,J)+B(I,J)                                            MADD   12
C                                                                       MADD   13
C                                                                       MADD   14
      RETURN                                                            MADD   15
      END                                                               MADD   16
```

```
      SUBROUTINE MREAD (A,N,M)                                          MRED    1
C     THIS SUBROUTINE READS AN NXM MATRIX A ACCORDING TO THE FORMAT     MRED    2
C     8F10.0. THE ENTRIES IN THE FIRST ROW OF A ARE READ FIRST, THEN    MRED    3
C     THE ENTRIES IN THE SECOND ROW, AND SO ON.                         MRED    4
      REAL *8A                                                          MRED    5
      DIMENSION A(9,9)                                                  MRED    6
C                                                                       MRED    7
C                                                                       MRED    8
      DO 1 I=1,N                                                        MRED    9
    1 READ (5,2) (A(I,J),J=1,M)                                         MRED   10
C                                                                       MRED   11
C                                                                       MRED   12
      RETURN                                                            MRED   13
C                                                                       MRED   14
C                                                                       MRED   15
    2 FORMAT (8F10.0)                                                   MRED   16
      END                                                               MRED   17
```

```
      SUBROUTINE MWRITE (A,N,M)                                          MWRT    1
C     THIS SUBROUTINE WRITES THE ENTRIES OF THE NXM MATRIX A             MWRT    2
      REAL *8A                                                           MWRT    3
      DIMENSION A(9,9)                                                   MWRT    4
C                                                                        MWRT    5
CC                                                                       MWRT    6
      DO 1 I=1,N                                                         MWRT    7
    1 WRITE (6,2) (A(I,J),J=1,M)                                         MWRT    8
C                                                                        MWRT    9
CC                                                                       MWRT   10
      RETURN                                                             MWRT   11
C                                                                        MWRT   12
CC                                                                       MWRT   13
    2 FORMAT (9(2X,1PE12.5))                                             MWRT   14
      END                                                                MWRT   15
```

```
      SUBROUTINE PROD (A,B,N,M,L,C)                                    MPRD    1
C     THIS SUBROUTINE COMPUTES THE MATRIX PRODUCT AB AND STORES THE    MPRD    2
C     RESULT IN C                                                      MPRD    3
C     A = NXM,    B = MXL,    C = NXL                                  MPRD    4
      REAL *8A,B,C,T                                                   MPRD    5
      DIMENSION A(9,9), B(9,9), C(9,9), T(9,9)                         MPRD    6
C                                                                      MPRD    7
C                                                                      MPRD    8
      DO 1 I=1,N                                                       MPRD    9
C                                                                      MPRD   10
C                                                                      MPRD   11
      DO 1 J=1,L                                                       MPRD   12
    1 T(I,J) = 0.0                                                     MPRD   13
C                                                                      MPRD   14
CCCC                                                                   MPRD   15
C                                                                      MPRD   16
      DO 2 I=1,N                                                       MPRD   17
C                                                                      MPRD   18
C                                                                      MPRD   19
      DO 2 J=1,L                                                       MPRD   20
C                                                                      MPRD   21
C                                                                      MPRD   22
      DO 2 K=1,M                                                       MPRD   23
    2 T(I,J) = T(I,J)+A(I,K)*B(K,J)                                    MPRD   24
C                                                                      MPRD   25
CCCC                                                                   MPRD   26
C                                                                      MPRD   27
      DO 3 I=1,N                                                       MPRD   28
C                                                                      MPRD   29
C                                                                      MPRD   30
      DO 3 J=1,L                                                       MPRD   31
    3 C(I,J) = T(I,J)                                                  MPRD   32
C                                                                      MPRD   33
C                                                                      MPRD   34
      RETURN                                                           MPRD   35
      END                                                              MPRD   36
                                                                       MPRD   37
                                                                       MPRD   38
```

```
      SUBROUTINE SUB (A,B,N,M,C)                                        MSUB    1
C     THIS SUBROUTINE SUBTRACTS THE NXM MATRIX B FROM THE NXM MATRIX    MSUB    2
C     A AND STORES THE RESULT IN C                                      MSUB    3
      REAL *8A,B,C                                                      MSUB    4
      DIMENSION A(9,9), B(9,9), C(9,9)                                  MSUB    5
C                                                                       MSUB    6
C                                                                       MSUB    7
      DO 1 I=1,N                                                        MSUB    8
C                                                                       MSUB    9
C                                                                       MSUB   10
      DO 1 J=1,M                                                        MSUB   11
    1 C(I,J) = A(I,J)-B(I,J)                                            MSUB   12
C                                                                       MSUB   13
C                                                                       MSUB   14
      RETURN                                                            MSUB   15
      END                                                               MSUB   16
```

```
      SUBROUTINE TRANS (A,N,M,C)                                        MTRN    1
C     THIS SUBROUTINE FORMS THE MATRIX TRANSPOSE OF A STORING THE       MTRN    2
C     RESULT IN C                                                       MTRN    3
C     A = NXM,    C = MXN                                               MTRN    4
      REAL *8A,C                                                        MTRN    5
      DIMENSION A(9,9), C(9,9)                                          MTRN    6
C                                                                       MTRN    7
C                                                                       MTRN    8
      DO 1 I=1,N                                                        MTRN    9
C                                                                       MTRN   10
C                                                                       MTRN   11
      DO 1 J=1,M                                                        MTRN   12
    1 C(J,I) = A(I,J)                                                   MTRN   13
C                                                                       MTRN   14
C                                                                       MTRN   15
      RETURN                                                            MTRN   16
      END                                                               MTRN   17
```

```
      SUBROUTINE VADD (X,Y,N,Z)                                          VADD    1
C     THIS SUBROUTINE COMPUTES THE SUM OF THE N-VECTORS X AND            VADD    2
C     Y AND STORES THE RESULT IN THE N-VECTOR Z                          VADD    3
C                                                                        VADD    4
      REAL *4X(9),Y(9),Z(9)                                              VADD    5
C                                                                        VADD    6
C                                                                        VADD    7
      DO 1 I=1,N                                                         VADD    8
    1 Z(I) = X(I)+Y(I)                                                   VADD    9
C                                                                        VADD   10
C                                                                        VADD   11
      RETURN                                                             VADD   12
      END                                                                VADD   13
```

```
      SUBROUTINE VPROD (A,X,M,N,Y)                                      VPRD    1
C     THIS SUBROUTINE COMPUTES THE PRODUCT OF THE MXN MATRIX            VPRD    2
C     A AND THE N-VECTOR X AND STORES THE RESULT IN THE                 VPRD    3
C     M-VECTOR Y                                                        VPRD    4
C                                                                       VPRD    5
C                                                                       VPRD    6
      REAL *4A(9,9),X(9),Y(9),T(9)                                      VPRD    7
C                                                                       VPRD    8
C                                                                       VPRD    9
      DO 1 I=1,M                                                        VPRD   10
      T(I) = 0.DO                                                       VPRD   11
C                                                                       VPRD   12
C                                                                       VPRD   13
      DO 1 J=1,N                                                        VPRD   14
    1 T(I) = T(I)+A(I,J)*X(J)                                           VPRD   15
C                                                                       VPRD   16
CC                                                                      VPRD   17
CC                                                                      VPRD   18
C                                                                       VPRD   19
      DO 2 I=1,M                                                        VPRD   20
    2 Y(I) = T(I)                                                       VPRD   21
C                                                                       VPRD   22
C                                                                       VPRD   23
      RETURN                                                            VPRD   24
      END                                                               VPRD   25
```

```
      SUBROUTINE VREAD (V,N)                                          VRED    1
C                                                                     VRED    2
C     THIS SUBROUTINE READS THE N-DIMENSIONAL S.P. VECTOR V           VRED    3
C                                                                     VRED    4
      DIMENSION V(9)                                                  VRED    5
      READ (5,1) (V(I),I=1,N)                                         VRED    6
      RETURN                                                          VRED    7
C                                                                     VRED    8
C                                                                     VRED    9
    1 FORMAT (8F10.0)                                                 VRED   10
      END                                                             VRED   11
```

```
      SUBROUTINE VSUB (X,Y,N,Z)                                          VSUB    1
C     THIS SUBROUTINE COMPUTES THE DIFFERENCE X-Y OF THE TWO             VSUB    2
C     N-VECTORS X & Y AND STORES THE RESULT IN THE N-VECTOR Z            VSUB    3
C                                                                        VSUB    4
      REAL *4X(9),Y(9),Z(9)                                              VSUB    5
C                                                                        VSUB    6
C                                                                        VSUB    7
      DO 1 I=1,N                                                         VSUB    8
    1 Z(I) = X(I)-Y(I)                                                   VSUB    9
C                                                                        VSUB   10
C                                                                        VSUB   11
      RETURN                                                             VSUB   12
      END                                                                VSUB   13
```

```
      SUBROUTINE VWRITE (V,N)                                          VWRT    1
C                                                                      VWRT    2
CC    THIS SUBROUTINE WRITES THE N-DIMENSIONAL S.P. VECTOR V           VWRT    3
C                                                                      VWRT    4
      DIMENSION V(9)                                                   VWRT    5
      WRITE (6,1) (V(I),I=1,N)                                         VWRT    6
      RETURN                                                           VWRT    7
C                                                                      VWRT    8
C                                                                      VWRT    9
    1 FORMAT (9(2X,1PE12.5))                                           VWRT   10
      END                                                              VWRT   11
```

A.2 LISTING OF KEY VARIABLES USED IN THE MONTE CARLO PROGRAM

AMISS	average number of misses over the entire track (scalar).
COVW(m,m)	covariance of the system random forcing input matrix (double precision).
EI(n,n)	identity matrix (double precision).
ERR(N,nsam)	sample mean of estimation error vector (single precision).
G(n,m)	sample gain matrix (double precision).
GAMMA(n,m)	system random forcing input distribution matrix (double precision).
GAMMAS(n,m)	system random forcing input distribution matrix (single precision).
GK(n,m)	sample gain matrix (single precision).
GKS(n,m,nsam)	filter gain schedule (single precision).
H(m,n)	system observation matrix (double precision).
HIT	average number of hits over the entire track (scalar).
HS(m,n)	system observation matrix (single precision).
ITER	ensemble member index.
M	number of measurements.
MISSD(nens,nsam)	sample miss distance matrix (double precision).
N	system order.
NENS	number of members in the ensemble.
NSAM	number of time points to be processed.
PHI(n,n)	state transition matrix (double precision).

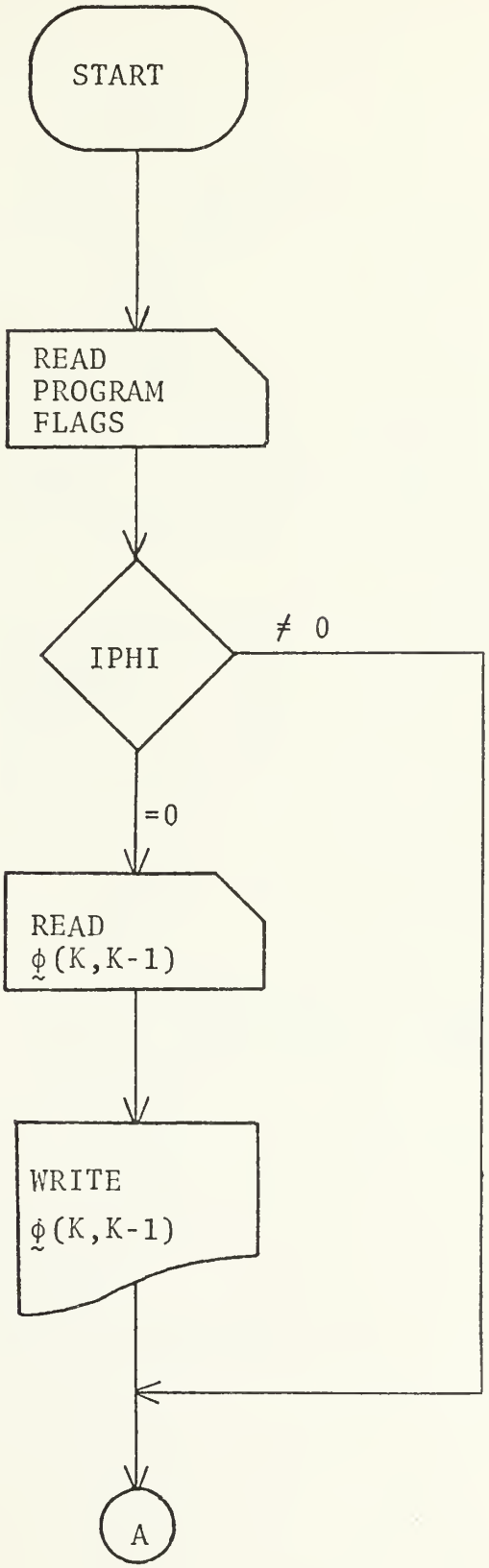
PHIS(n,n)	state transition matrix (single precision).
PKK(n,n)	theoretical covariance of estimation error matrix $\underset{\sim}{P}(K/K)$ (double precision).
PKKM1(n,n)	one-step prediction error covariance matrix $\underset{\sim}{P}(K+1/K)$ (double precision).
PKKM10(n,n)	initial value of the covariance of estimation error matrix $\underset{\sim}{P}(0/-1)$ (double precision).
PKKS(n,n,nsam)	sample covariance of estimation error matrix $\underset{\sim}{P}(K/K)$ (single precision).
Q(m,m)	system random forcing input covariance matrix (double precision).
R(m,m)	covariance of measurement noise matrix (double precision).
SCARE	average number of scares over the entire track.
SIGV(m)	standard deviation of measurement noise vector (single precision).
SIGW(m)	standard deviation of system random forcing input vector (single precision).
SIGXZ(n)	standard deviation of system initial condition vector (single precision).
V(m)	measurement noise vector (single precision).
VAR(n,n,nsam)	actual covariance of estimation error matrix (single precision).
W(m)	system random forcing input vector (single precision).
X(n)	state vector (single precision).
XHATZ(n)	initializing value of the state estimate vector $\underset{\sim}{\hat{x}}(0/-1)$ (single precision).
XHKK(n)	state estimate vector $\underset{\sim}{\hat{x}}(K/K)$ (single precision).
XHKKM1(n)	one-step state prediction vector $\underset{\sim}{\hat{x}}(K/K-1)$ (single precision).

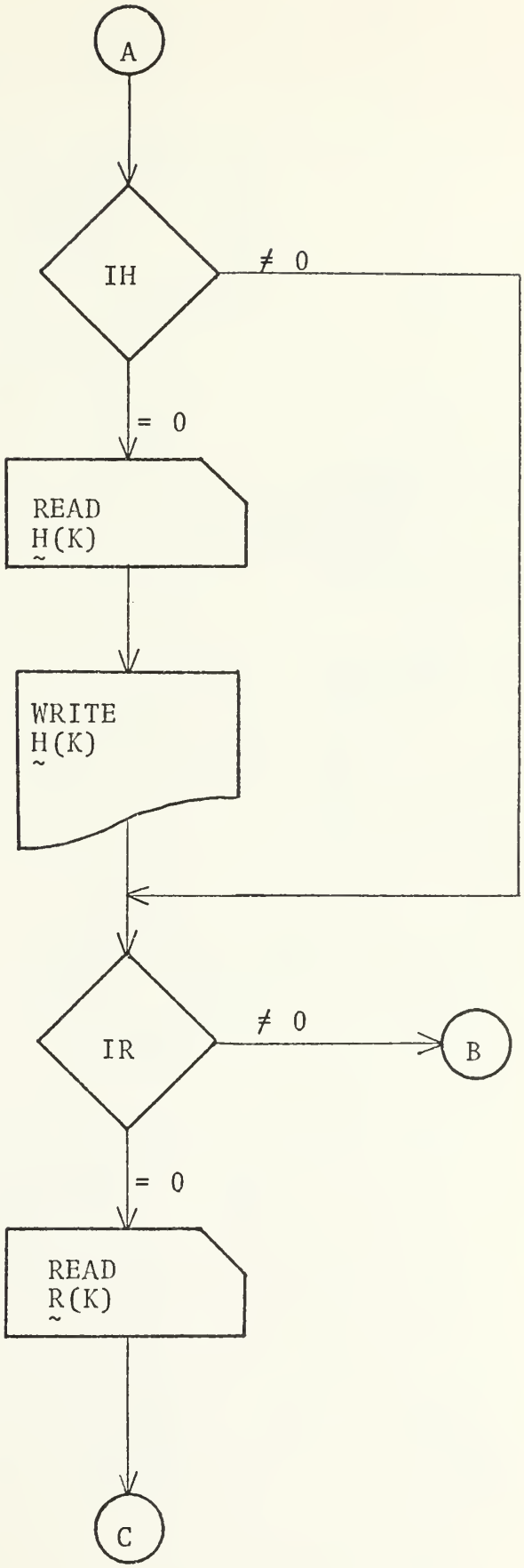
XM(n,nsam)	mean state trajectory over the ensemble (single precision).
XS(n,nsam)	reference trajectory (single precision).
XZMEAN(n)	mean of initial condition vector (single precision).

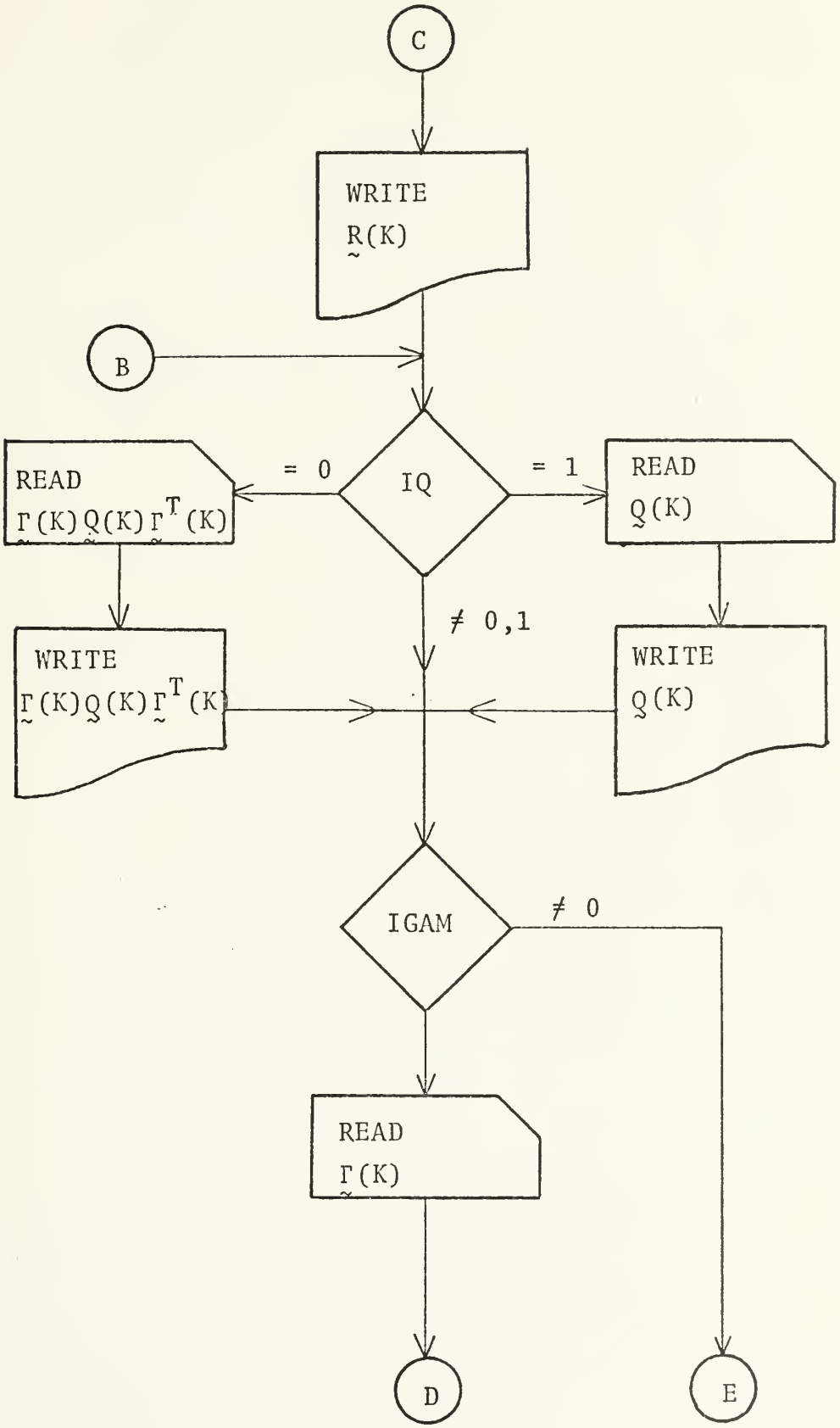
A.3 IBM SYSTEM 360/67 SOURCE LIBRARY SUBROUTINES CALLED BY THE MONTE CARLO SIMULATION PROGRAM

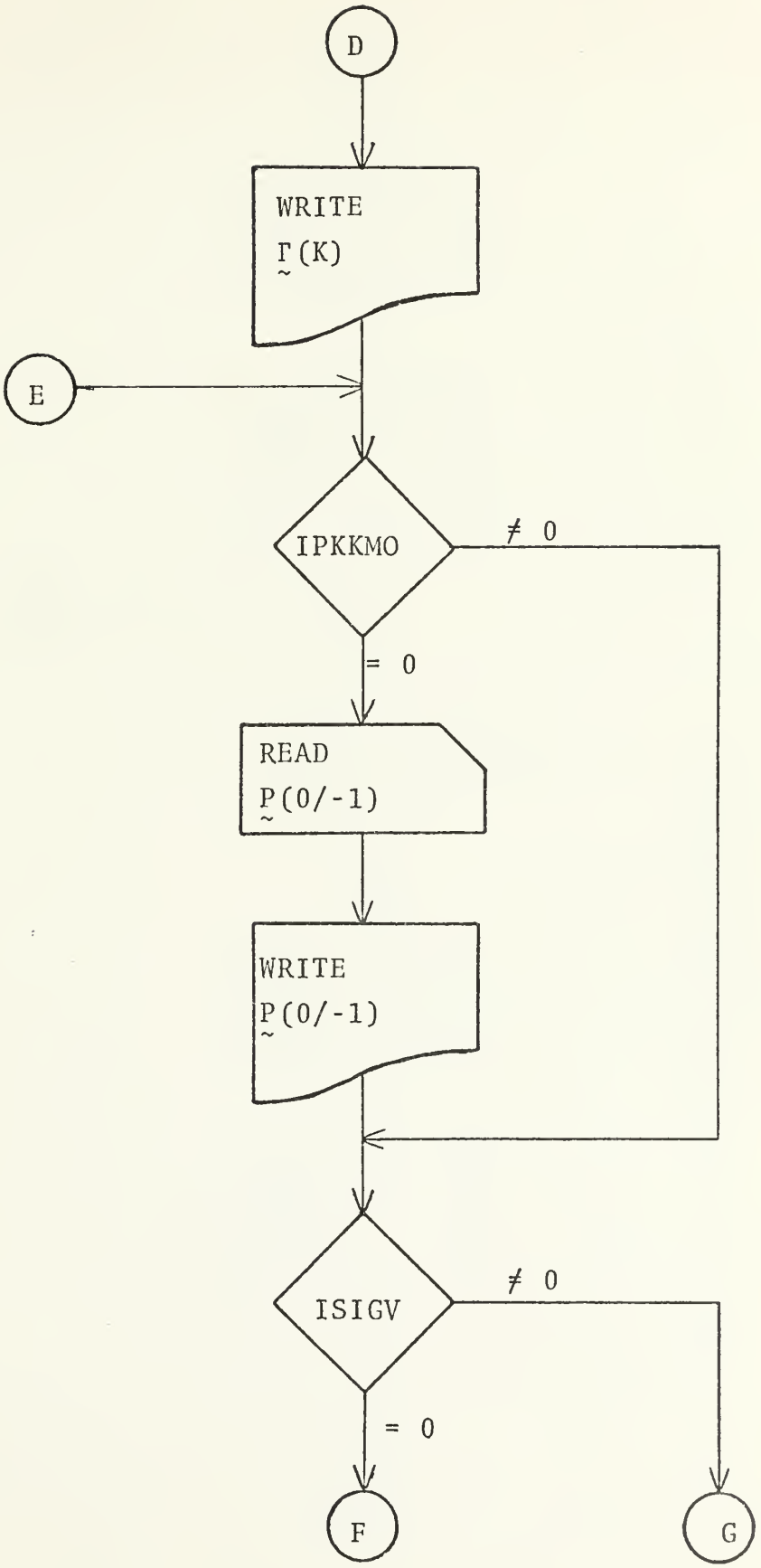
GAUSS3	matrix inversion by the method of GAUSS.
OVFLOW	utility program required by the random number generator.
PLOTP	generates two-dimensional line printer plots.
SNORM	random number generator that gives normally distributed, shuffled pseudo-random numbers.

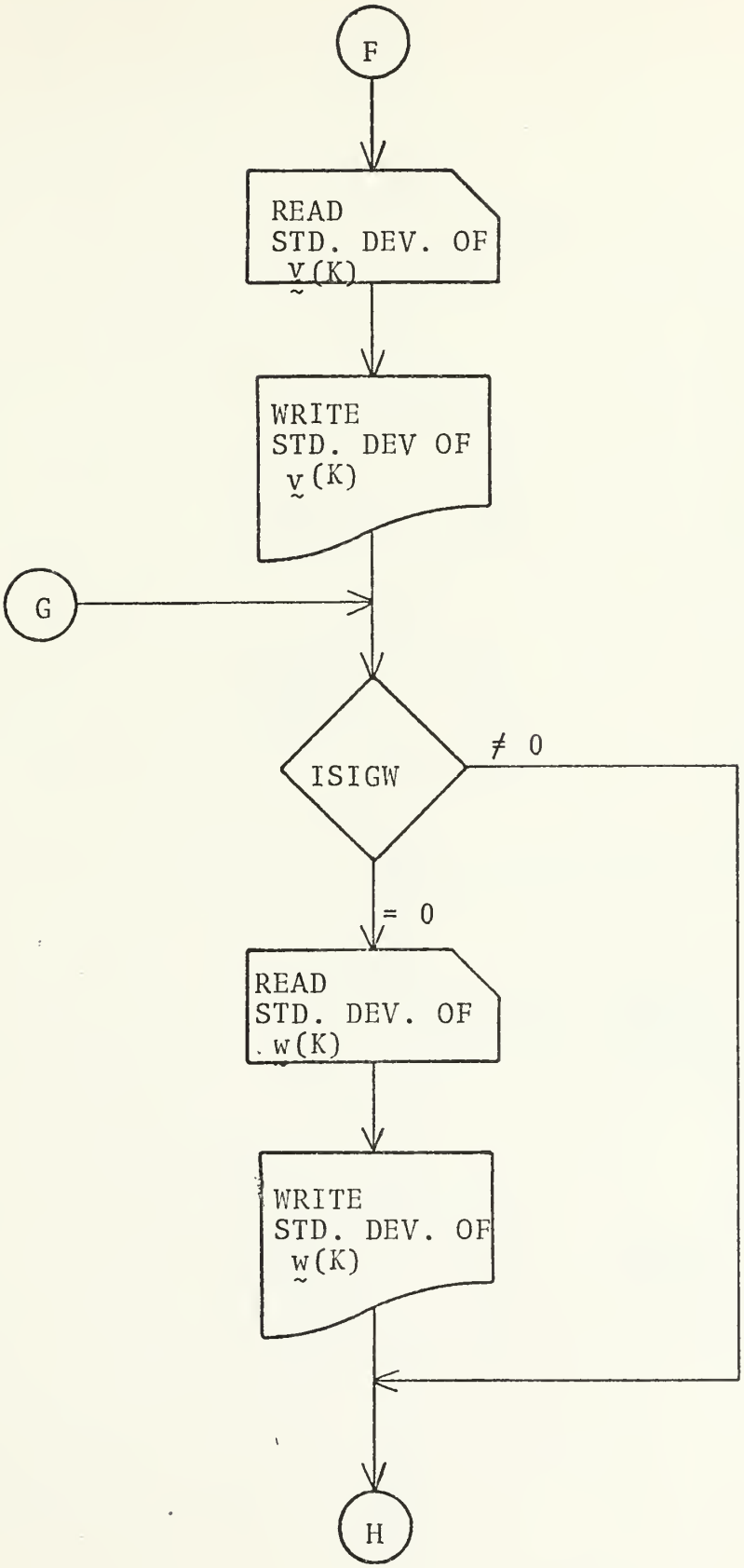
A.4 MONTE CARLO SIMULATION PROGRAM FLOWCHART

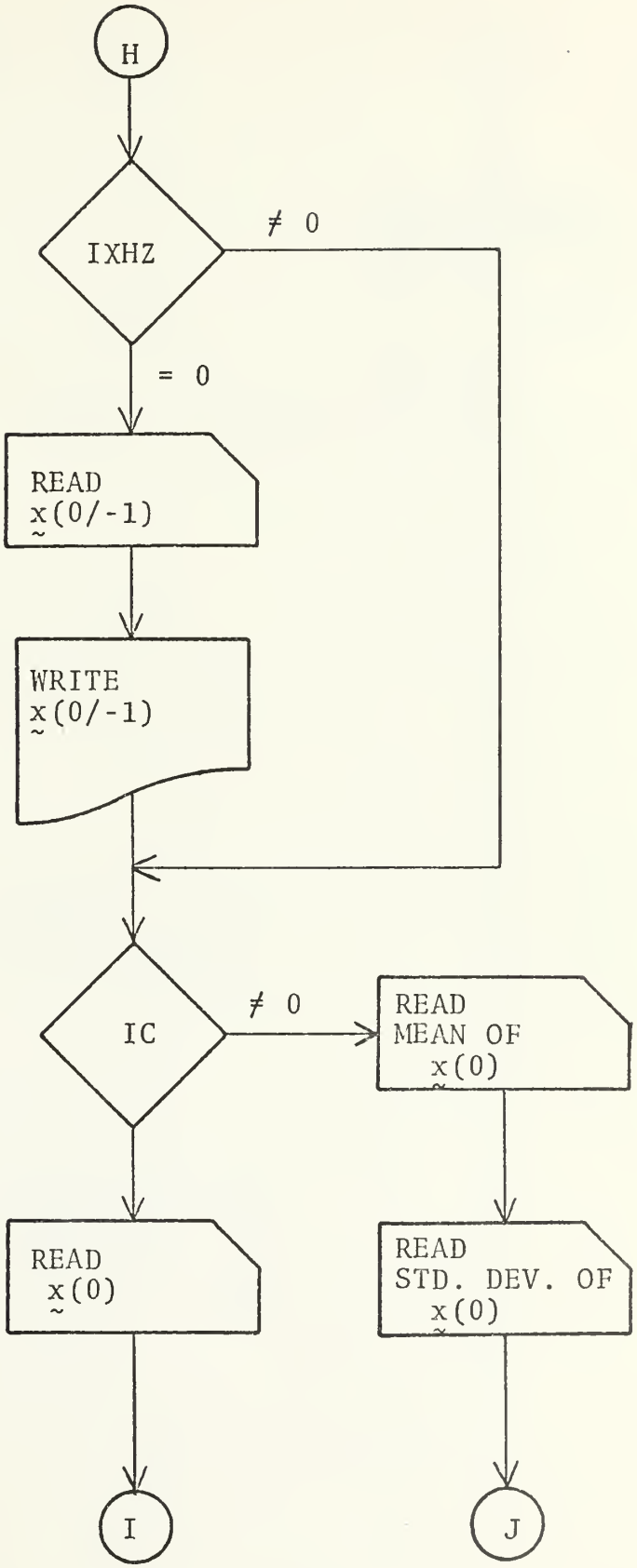


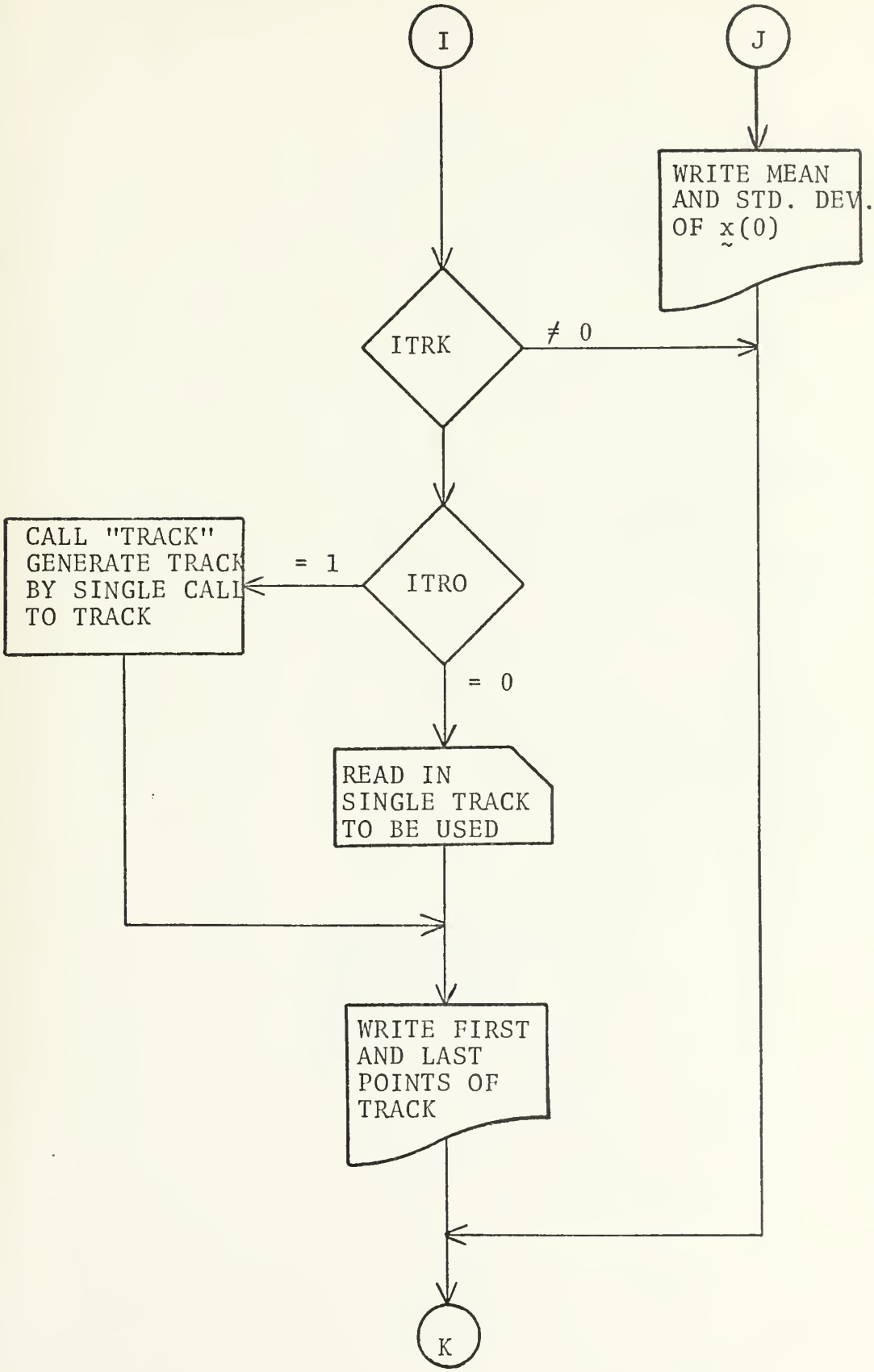

A
IH
≠ 0
= 0
READ
H(K)
~
WRITE
H(K)
~
IR
≠ 0
B
= 0
READ
R(K)
~
C

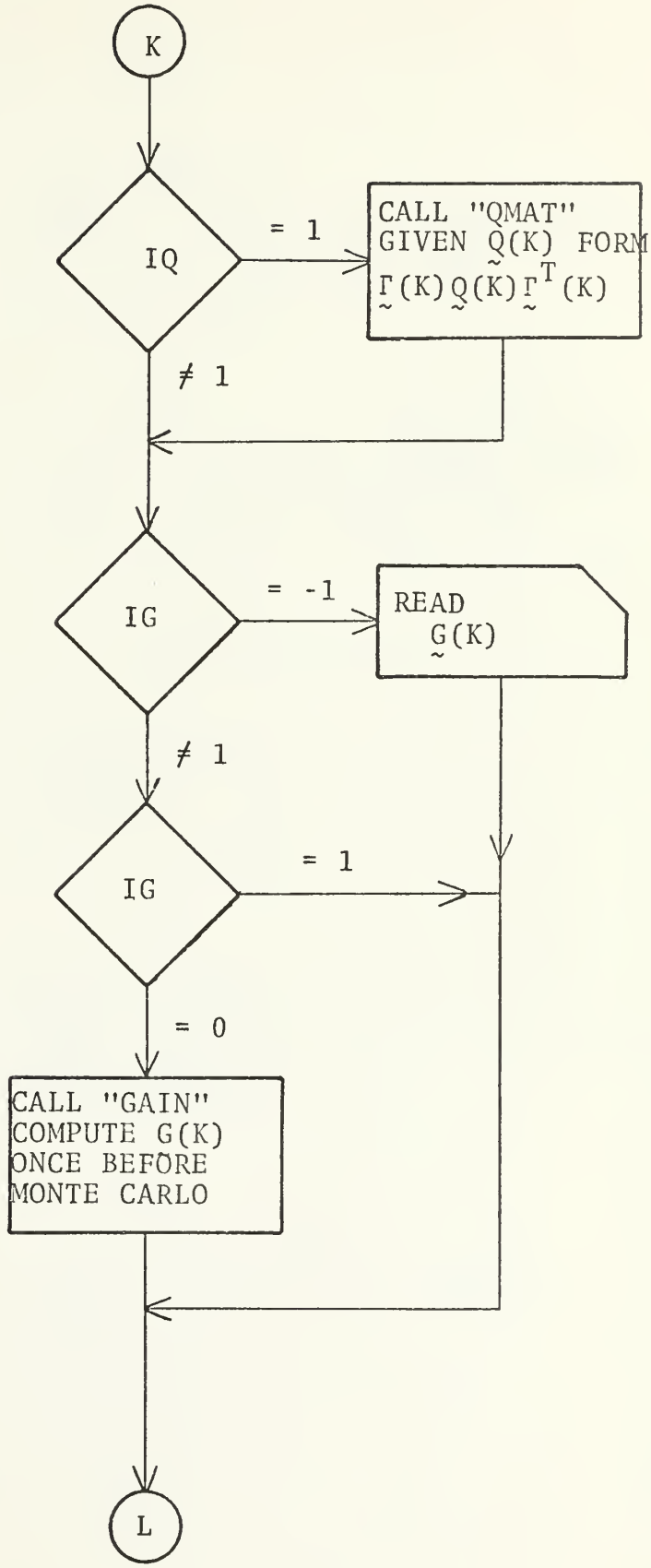

C
WRITE
R(K)
B
IQ
= 0
= 1
READ
Γ(K)Q(K)Γ^T(K)
READ
Q(K)
≠ 0,1
WRITE
Γ(K)Q(K)Γ^T(K)
WRITE
Q(K)
IGAM
≠ 0
READ
Γ(K)
D
E

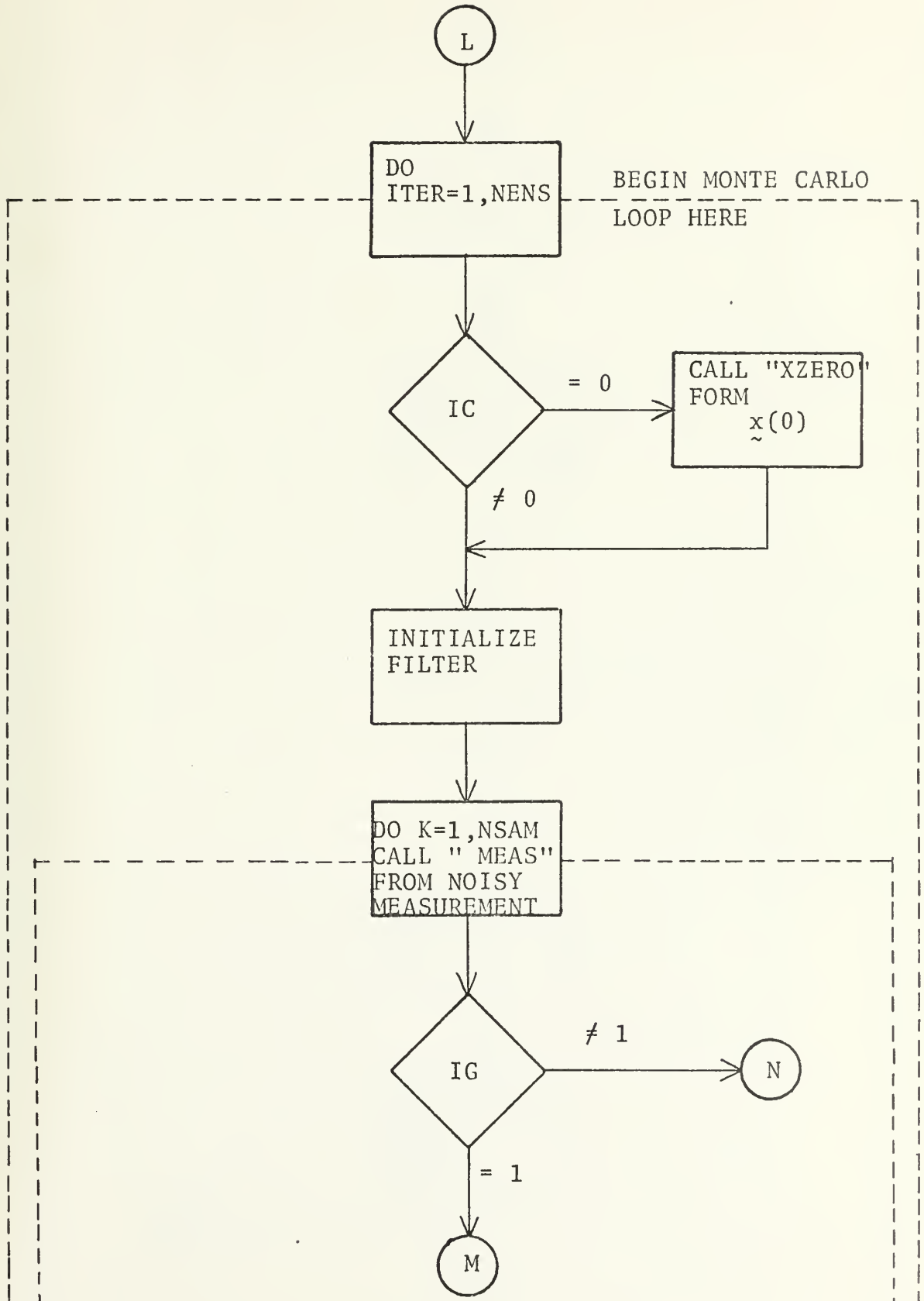

D
WRITE
Γ(K)
E
IPKKMO
≠ 0
= 0
READ
P(0/-1)
WRITE
P(0/-1)
ISIGV
≠ 0
= 0
F
G

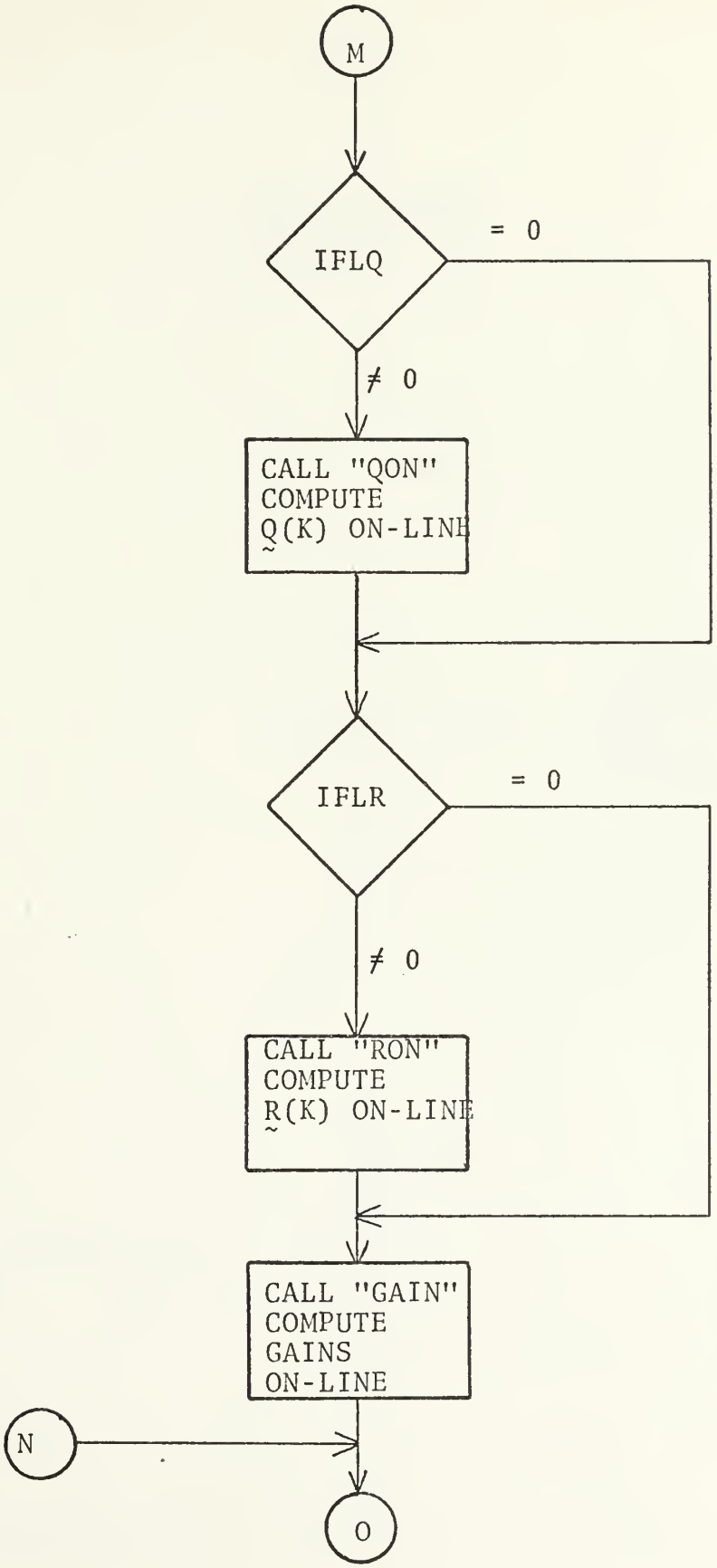

F
READ
STD. DEV. OF
v(K)
WRITE
STD. DEV OF
v(K)
G
ISIGW
≠ 0
= 0
READ
STD. DEV. OF
w(K)
WRITE
STD. DEV. OF
w(K)
H

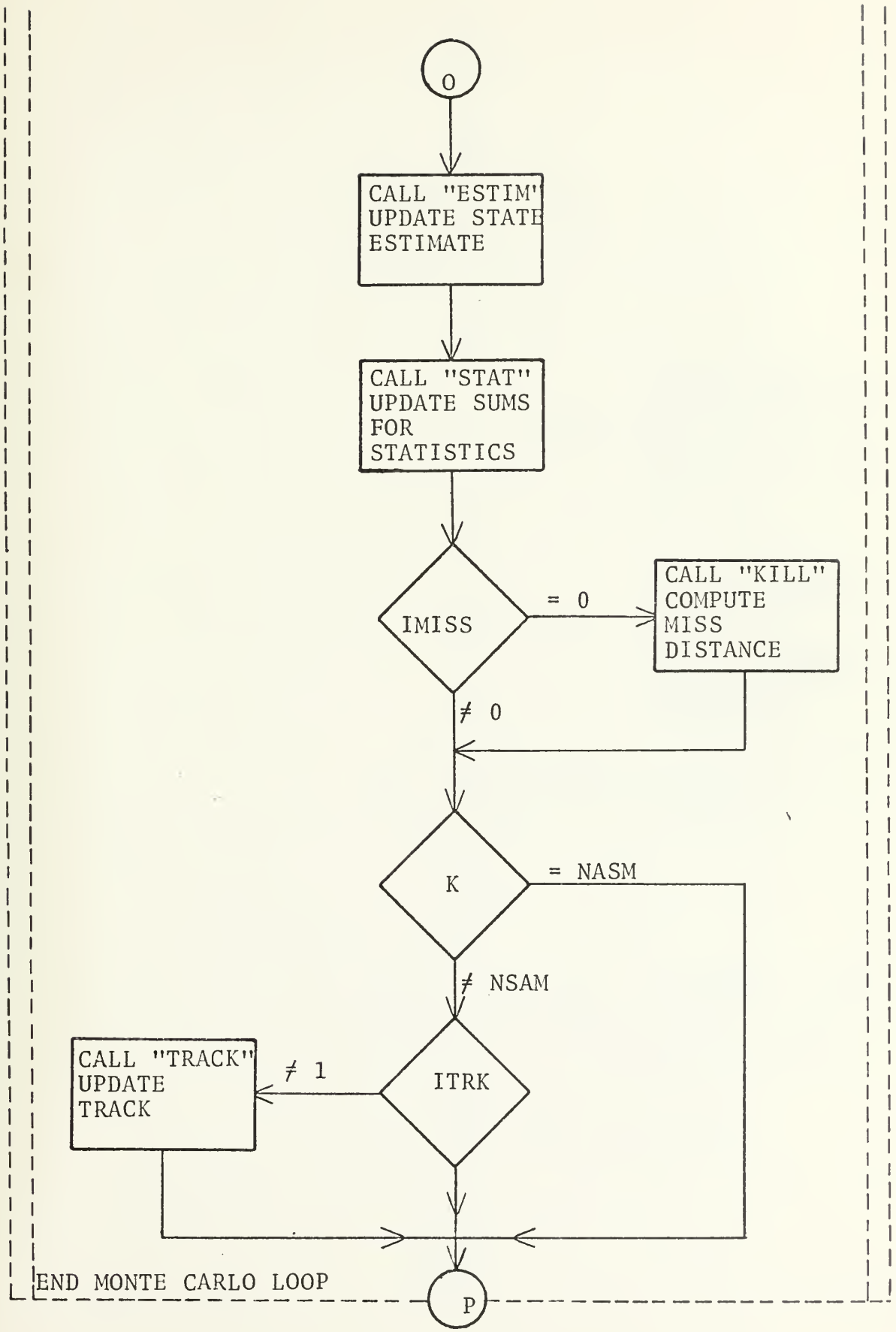

H
IXHZ
≠ 0
= 0
READ
x(0/-1)
WRITE
x(0/-1)
IC
≠ 0
READ
MEAN OF
x(0)
READ
x(0)
READ
STD. DEV. OF
x(0)
I
J

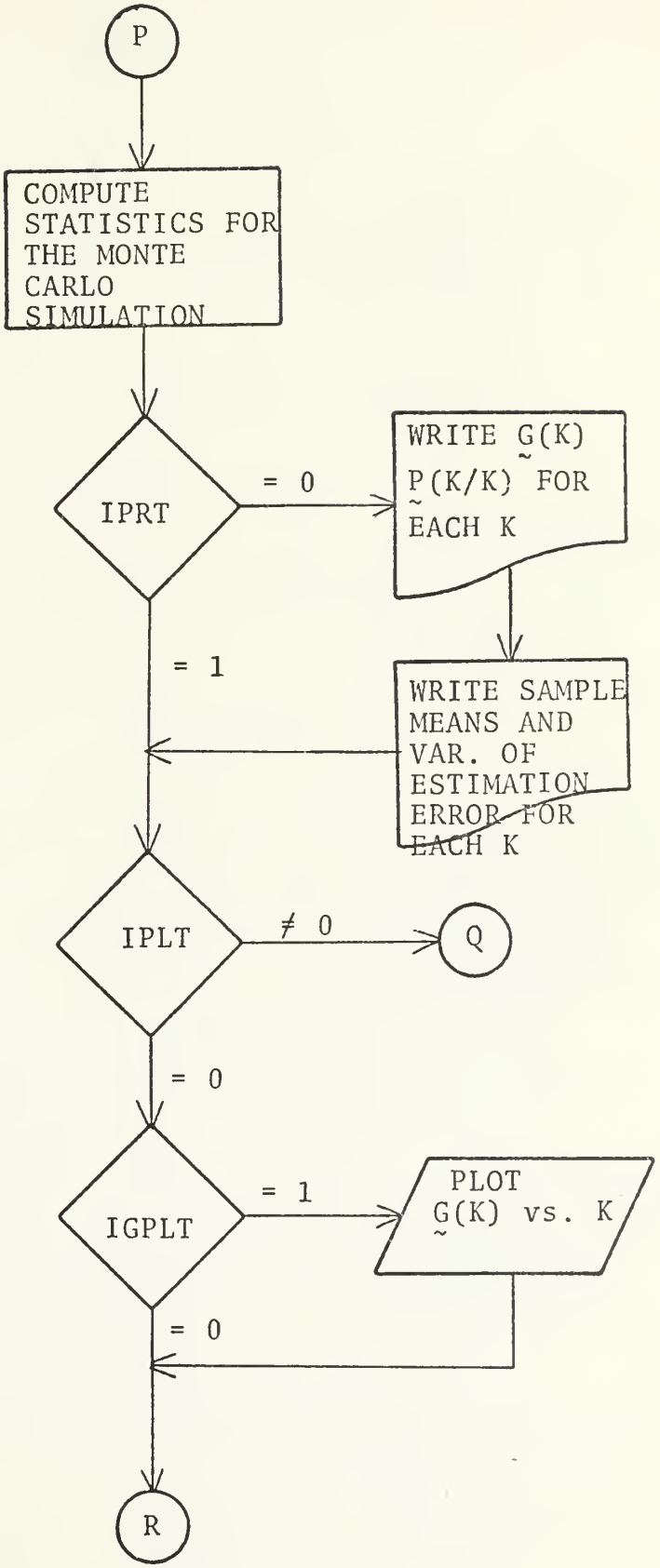

I
J
WRITE MEAN AND STD. DEV. OF x̃(0)
ITRK
≠ 0
CALL "TRACK" GENERATE TRACK BY SINGLE CALL TO TRACK
= 1
ITRO
= 0
READ IN SINGLE TRACK TO BE USED
WRITE FIRST AND LAST POINTS OF TRACK
K

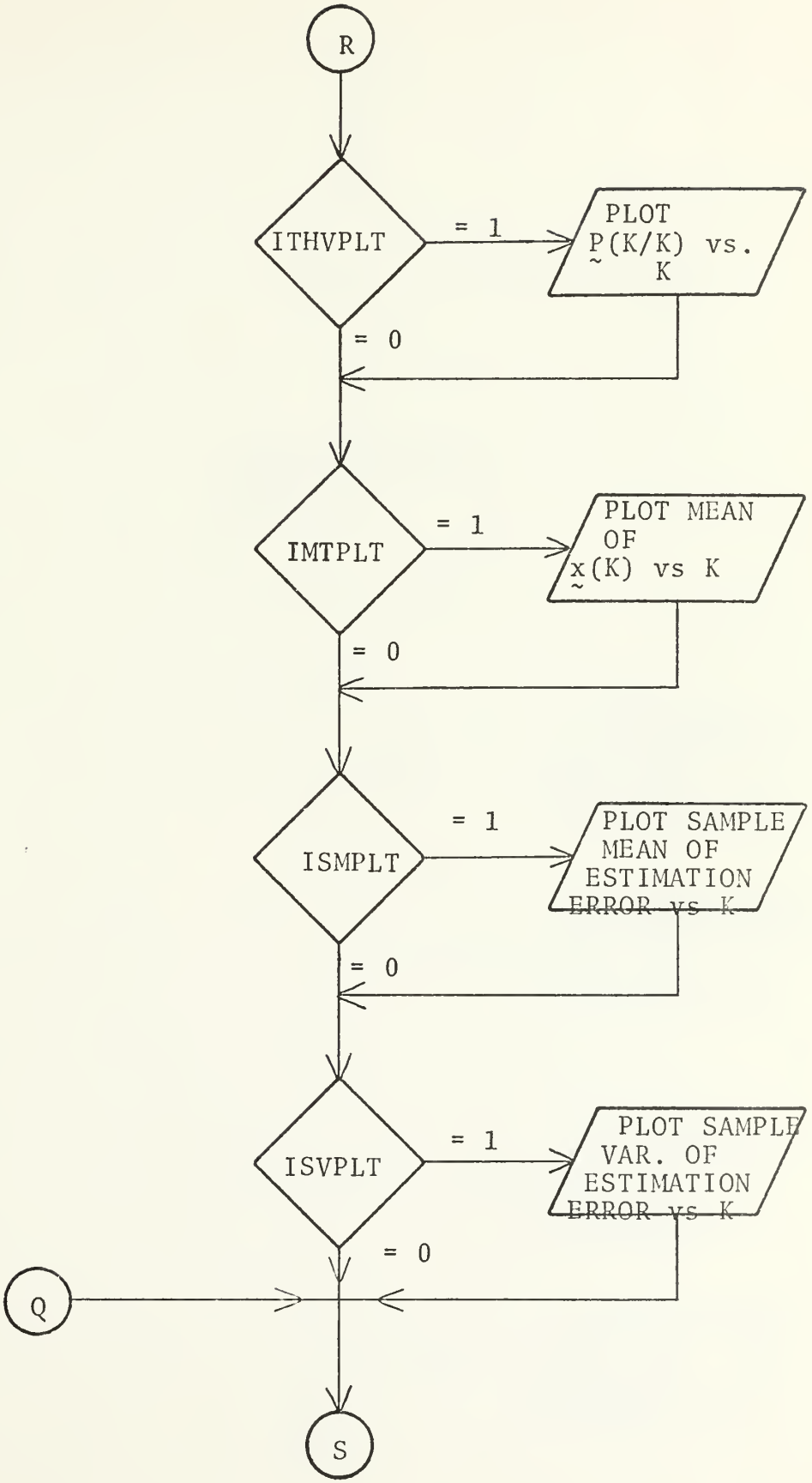

K
IQ
= 1
CALL "QMAT"
GIVEN Q̃(K) FORM
Γ̃(K) Q̃(K) Γ̃^T(K)
≠ 1
IG
= -1
READ
G̃(K)
≠ 1
IG
= 1
= 0
CALL "GAIN"
COMPUTE G(K)
ONCE BEFORE
MONTE CARLO
L

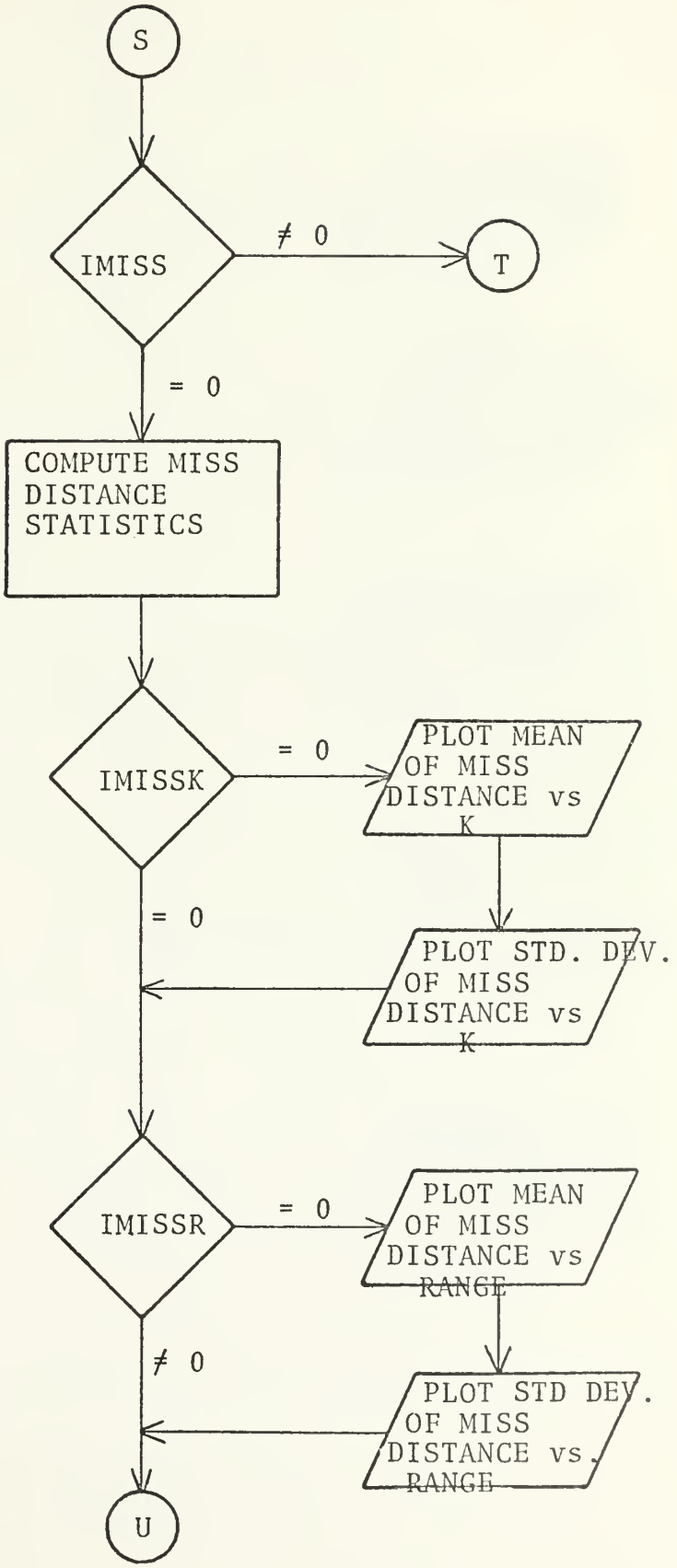

L
DO
ITER=1,NENS
BEGIN MONTE CARLO
LOOP HERE
IC
= 0
CALL "XZERO"
FORM
x(0)
≠ 0
INITIALIZE
FILTER
DO K=1,NSAM
CALL " MEAS"
FROM NOISY
MEASUREMENT
IG
≠ 1
N
= 1
M

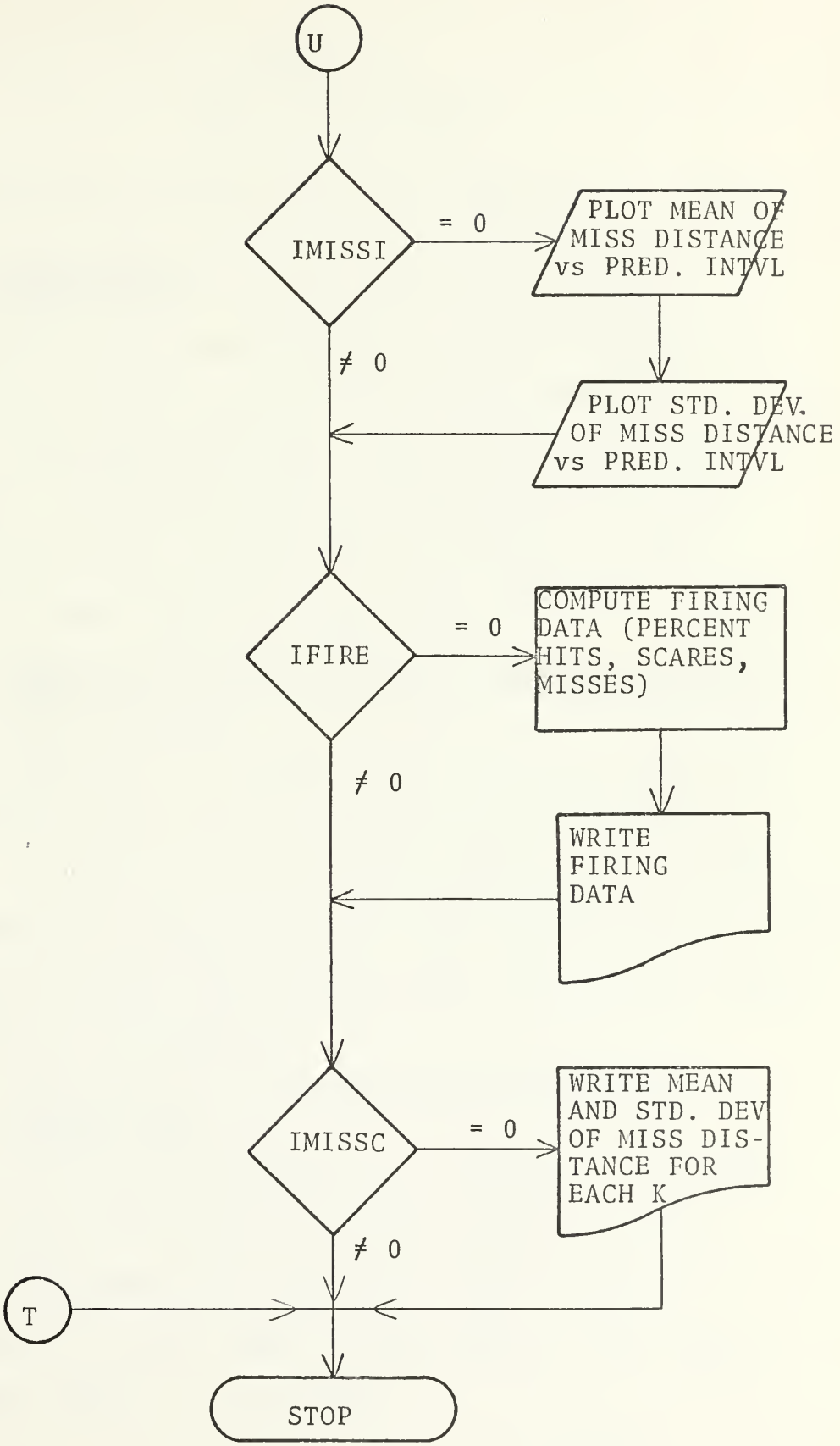

M
IFLQ
= 0
≠ 0
CALL "QON"
COMPUTE
Q(K) ON-LINE
IFLR
= 0
≠ 0
CALL "RON"
COMPUTE
R(K) ON-LINE
CALL "GAIN"
COMPUTE
GAINS
ON-LINE
N
O


O
CALL "ESTIM"
UPDATE STATE
ESTIMATE
CALL "STAT"
UPDATE SUMS
FOR
STATISTICS
IMISS
= 0
CALL "KILL"
COMPUTE
MISS
DISTANCE
≠ 0
K
= NASM
≠ NSAM
ITRK
≠ 1
CALL "TRACK"
UPDATE
TRACK
END MONTE CARLO LOOP
P


P
COMPUTE STATISTICS FOR THE MONTE CARLO SIMULATION
IPRT
= 0
WRITE G(K) P(K/K) FOR EACH K
= 1
WRITE SAMPLE MEANS AND VAR. OF ESTIMATION ERROR FOR EACH K
IPLT
≠ 0
Q
= 0
IGPLT
= 1
PLOT G(K) vs. K
= 0
R


R
ITHVPLT
= 1
PLOT
P(K/K) vs.
K
= 0
IMTPLT
= 1
PLOT MEAN
OF
x(K) vs K
= 0
ISMPLT
= 1
PLOT SAMPLE
MEAN OF
ESTIMATION
ERROR vs K
= 0
ISVPLT
= 1
PLOT SAMPLE
VAR. OF
ESTIMATION
ERROR vs K
= 0
Q
S


S
IMISS
≠ 0
T
= 0
COMPUTE MISS DISTANCE STATISTICS
IMISSK
= 0
PLOT MEAN OF MISS DISTANCE vs K
= 0
PLOT STD. DEV. OF MISS DISTANCE vs K
IMISSR
= 0
PLOT MEAN OF MISS DISTANCE vs RANGE
≠ 0
PLOT STD DEV. OF MISS DISTANCE vs. RANGE
U


U
IMISSI
= 0
PLOT MEAN OF MISS DISTANCE vs PRED. INTVL
≠ 0
PLOT STD. DEV. OF MISS DISTANCE vs PRED. INTVL
IFIRE
= 0
COMPUTE FIRING DATA (PERCENT HITS, SCARES, MISSES)
≠ 0
WRITE FIRING DATA
IMISSC
= 0
WRITE MEAN AND STD. DEV OF MISS DISTANCE FOR EACH K
≠ 0
T
STOP

APPENDIX B

TEST TRAJECTORIES

B.1 DESCRIPTION OF TEST TRAJECTORY PROFILES

a. Trajectory 1; short-range, low speed, non-maneuvering

Trajectory 1 commences at a range of 3850 yards in a 30 degree dive with an initial velocity of -230 yds/sec (the negative velocity indicates motion toward the origin). The trajectory is terminated at a final range of 457 yards. No maneuver is executed. Sixty data points were taken at a 4-Hz rate.

b. Trajectory 2; short-range, low speed, moderate level maneuver

Trajectory 2 commences at a range of 3850 yards in a 30 degree dive with an initial velocity of -230 yds/sec. At 3450 yards the target executes a 2.0 g $\sin^2$ planar maneuver lasting 13.5 sec. The maneuver is completed at the final sample range of 465 yards. Sixty data points were taken at a 4-Hz rate.

c. Trajectory 3; long-range, low speed, non-maneuvering

Trajectory 3 commences at a range of 9225 yards in a 30 degree dive with an initial velocity of -230 yds/sec. No maneuver is executed. The trajectory is terminated at a final range of 5832 yards. Sixty data points were taken at a 4-Hz rate.

d. Trajectory 4; long-range, low speed, moderate level maneuver

Trajectory 4 commences at a range of 9225 yards in a 30 degree dive with an initial velocity of -230 yds/sec. At 8822 yards, the target executes a 2.0 g $\sin^2$ planar maneuver lasting 13.5 sec. The maneuver is completed at the final sample range of 5840 yards. Sixty data points were taken at a 4-Hz rate.

e. Trajectory 5; medium-range, high speed, evasive level maneuver

Trajectory 5 commences at a range of 4500 yards in a 20 degree dive with an initial velocity of -300 yds/sec. At 3900 yards the target executes a 6.0 g $\sin^2$ planar maneuver lasting 10 sec. The maneuver is completed at a range of 914 yards and the trajectory is terminated at a final range of 90 yards. Sixty data points were taken at a 4-Hz rate.

f. Trajectory 6; long-range, high speed, evasive level maneuver

Trajectory 6 commences at a range of 8000 yards in a 20 degree dive with an initial velocity of -300 yds/sec. At 7500 yards the target executes a 6.0 g $\sin^2$ planar maneuver lasting 10 sec. The maneuver is completed at a range of 4340 and the trajectory is terminated at a range of 3590 yards. Sixty data points were taken at a 4-Hz rate.

g. Trajectory 7; medium-range, high speed, non-maneuvering

Trajectory 7 commences at a range of 5000 yards in a 20 degree dive with an initial velocity of -300 yds/sec. No maneuver is executed. The trajectory is terminated at a final range of 575 yards. Sixty data points were taken at a 4-Hz rate.

h. Trajectory 8; long-range, high speed, non-maneuvering

Trajectory 8 commences at a range of 8000 yards in a 20 degree dive with an initial velocity of -300 yds/sec. No maneuver is executed. The trajectory is terminated at a final range of 3575 yards. Sixty data points were taken at a 4-Hz rate.

i. Trajectory 9; long-range, low speed, non-maneuvering

Trajectory 9 commences at a range of 8000 yards in 20 degree dive with an initial velocity of -200 yds/sec. No maneuver is executed. The trajectory is terminated at the final sample range of 50 yards. One-hundred-sixty data points were taken at a 4-Hz rate.

j. Trajectory 10; long-range, low speed, moderate level maneuver

Trajectory 10 commences at a range of 8000 yards in a 20 degree dive with an initial velocity of -200 yds/sec. At 4000 yards the target executes a 2.0 g $\sin^2$ planar maneuver lasting 13.5 sec. The maneuver is completed at a range of 1200 yards and the trajectory is terminated at

the final sample range of 50 yards. One-hundred-sixty data points were taken at a 4-Hz rate.

k. Trajectory 11; long-range, low speed, moderate level maneuver

Trajectory 11 commences at a range of 8000 yards in a 20 degree dive with an initial velocity of -200 yds/sec. At 2700 yards the target executes a 2.0 g $\sin^2$ planar maneuver lasting 13.5 sec. The maneuver is completed at the final sample range of 50 yards. One-hundred-sixty data points were taken at a 4-Hz rate.

l. Trajectory 12; long-range, low speed, moderate level maneuver

Trajectory 12 commences at a range of 8000 yards in a 20 degree dive with an initial velocity of -200 yds/sec. At 3333 yards the target executes a 2.0 g $\sin^2$ planar maneuver lasting 13.5 sec. The maneuver is completed at a range of 500 yards and the trajectory is terminated at the final sample range of 50 yards. One-hundred-sixty data points were taken at a 4-Hz rate.

m. Trajectory 13; long-range, low speed, moderate level maneuver

Trajectory 13 commences at a range of 8000 yards in a 20 degree dive with an initial velocity of -200 yds/sec. At 2700 yards the target executes a 2.5 g $\sin^2$ planar maneuver lasting 10 sec. The maneuver is completed at the final sample range of 50 yards. One-hundred-sixty data points were taken at a 4-Hz rate.

n. Trajectory 14; long-range, low speed, moderate level maneuver

Trajectory 14 commences at a range of 8000 yards in a 20 degree dive with an initial velocity of -200 yds/sec. At 4000 yards the target executes a 2.5 g $\sin^2$ planar maneuver lasting 10 sec. The maneuver is completed at a range of 600 yards and the trajectory is terminated at the final sample range of 50 yards. One-hundred-sixty data points were taken at a 4-Hz rate.

B.2 TRAJECTORY GENERATING PROGRAM

The trajectories of Section B.1 are the products of a tracking generating program written by Parr. A complete description of its use, development of the planar $\sin^2$ maneuver, program listing and several examples may be found in Parr's work [Ref. 4].

APPENDIX C

THE N-STEP PREDICTION PROBLEM

The ultimate test of any fire control system is its ability to destroy an engaged target. To this end, the determining factor is the accuracy with which it predicts the future position of the target, since launcher laying orders are derived from knowledge of the target's present and future states. This of course, neglects the response characteristics of the launcher and the weapon (projectile) ballistics, both of which are assumed here to have negligible effects on the end result.

The interval over which the predictions are made is derived from a straight line approximation of the plot of projectile Time of Flight (TOF) vs. the target's present slant range, for a 5"/54 CAL. projectile. For the interval in question, $\{TOF: TOF\varepsilon(0.0,10.0)\}$ the straight line approximation takes on the value

$$TOF = 1.43 \times 10^{-3} \text{ (sec/yd)} \times \text{(slant range)} \quad \text{(C.1)}$$

The TOF interval, (0.0,10.0) was based on the assumption that the maximum effective range of the 5"/54 CAL gun against an air target is 7000 yds, corresponding to a TOF of approximately 10.0 seconds.

In the absence of any deterministic inputs to the system, the target's future position predicted over N samples is given by

$$\hat{\underset{\sim}{x}}(K+N/K) = \underset{\sim}{\phi}^{N}\hat{\underset{\sim}{x}}(K/K) \qquad (C.2)$$

Raising the state transition matrix $\underset{\sim}{\phi}(K,K+1)$ to a decimal power presents some computational difficulties, so N is formed by computing the TOF from $\hat{x}_1(K/K)$ (the current estimate of the target's slant range) then truncating this value to give the integer N. For large N the difficulty of raising a (9 x 9) matrix to the $N^{\underline{th}}$ power still exists. Note, however, that the assumption of linear plant dynamics decouples the filter in each of three coordinate directions. Partitioning and decoupling we have for the system,

$$\underset{\sim}{x}(K+1) = \begin{bmatrix} \underset{\sim}{\phi}_{11} & \underset{\sim}{0} & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{0}_{22} & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{0} & \underset{\sim}{0}_{33} \end{bmatrix} \begin{bmatrix} \underset{\sim}{x}_1(K) \\ \underset{\sim}{x}_2(K) \\ \underset{\sim}{x}_3(K) \end{bmatrix} + \begin{bmatrix} \underset{\sim}{\Gamma}_1 & \underset{\sim}{0} & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{\Gamma}_2 & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{0} & \underset{\sim}{\Gamma}_3 \end{bmatrix} \begin{bmatrix} \underset{\sim}{w}_1(K) \\ \underset{\sim}{w}_w(K) \\ \underset{\sim}{w}_3(K) \end{bmatrix} \qquad (C.3)$$

where

$$\underset{\sim}{x}_1(K) = [r,\dot{r},\ddot{r}]^T \qquad \underset{\sim}{w}_1(K) = [w_1(K)]$$

$$\underset{\sim}{x}_2(K) = [\phi,\dot{\phi},\ddot{\phi}]^T \qquad \underset{\sim}{w}_2(K) = [w_2(K)] \qquad (C.4)$$

$$\underset{\sim}{x}_3(K) = [\theta,\dot{\theta},\ddot{\theta}]^T \qquad \underset{\sim}{w}_3(K) = [w_3(K)]$$

Further, note that for each of the models discussed in this thesis, each of the $\underset{\sim}{\phi}_{ii}$ i = 1, 2, 3 are lower triangular matrices. Following the method given by Parr [Ref. 4], the elements of the $\underset{\sim}{\phi}_{ii}$ i = 1,2,3 are operated on individually and the resulting $\underset{\sim}{\phi}^{N}_{ii}$ i = 1,2,3 are adjoined to form the system state transition matrix $\underset{\sim}{\phi}^{N}(K+N,K)$. The intermediate result follows quickly and is shown on the following page.

$$\underset{\sim}{\phi}_{ii}^{N} = \begin{bmatrix} \phi_{ii}(1,1) & (\phi_{ii}(1,2))N & \left\{\phi_{ii}(1,3)\sum_{M=1}\phi_{ii}(3,3)^{M-1}\right. \\ & & \left. + \phi_{ii}(1,)\phi_{ii}(2,3)\sum_{M=1}^{N-1}(N-m)\phi_{ii}(3,3)^{M-1}\right\} \\ 0 & \phi_{ii}(2,2) & \phi_{ii}(2,3)\sum_{M=1}^{N}\phi_{ii}(3,3)^{M-1} \\ 0 & 0 & \phi_{ii}(3,3)^{N} \end{bmatrix} \qquad (C.5)$$

Finally, adjoining the $\underset{\sim}{\phi}^{N}_{ii}$'s gives

$$\underset{\sim}{\phi}^{N}(K+N,K) = \begin{bmatrix} \underset{\sim}{\phi}^{N}_{11} & \underset{\sim}{0} & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{\phi}^{N}_{22} & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{0} & \underset{\sim}{\phi}^{N}_{33} \end{bmatrix} \qquad (C.6)$$

The result shown in Equation (C.5) is easily programmed and involves simple manipulations without the need for extensive matrix multiplication. Note that the result shown in Equation (C.5) is equivalent to (for the single dimension)

$$r(K+N) = r(K) + \dot{r}(K)[(N-K)T] + \frac{1}{2}\ddot{r}(K)[(N-K)T]^2 \qquad (C.7)$$

For the case of the constant-velocity model, $\phi_{ii}(1,3)$, $\phi_{ii}(2,3)$, and $\phi_{ii}(3,j)$ $j = 1,2,3$ are set to zero. The prediction is then easily found by using Equation (C.2).

The magnitude of the N-step prediction error is defined to be the magnitude of a position vector, say $\overrightarrow{\text{ERRNET}}$, drawn between the estimate at time $t = (K+N)T$, $\hat{\underset{\sim}{x}}(K+N/K)$, and the reference trajectory, $\underset{\sim}{x}(K+N)$

$$||\overrightarrow{\text{ERRNET}}|| = ||\underset{\sim}{x}(K+N) - \hat{\underset{\sim}{x}}(K+N/K)|| \qquad (C.8)$$

Note that no reference is made to the direction of the error vector since the previously defined kill region is a sphere of constant radius centered at the reference position $\underset{\sim}{x}(K+N)$.

The primary interest lies in the following three regions

1) $||\overrightarrow{ERRNET}|| \leq 8.0$ yds — hit

2) $8.0 < ||\overrightarrow{ERRNET}|| \leq 16.0$ yds. — scare

3) $||\overrightarrow{ERRNET}|| > 16.0$ yds — miss

To avoid the necessity for coordinate conversion, the error vector is determined by a simple rotation of the polar plane followed by applications of the law of cosines. Consider the situation depicted in Figure 4, where points a and b represent $\hat{\underset{\sim}{x}}(K+N)$ and $\underset{\sim}{x}(K+N/K)$ (position only). Then from Equation (C.8)

$$||\overrightarrow{ERRNET}|| = ||\overrightarrow{ab}|| = ||\overrightarrow{ba}|| \qquad (C.9)$$

If the polar plane in Figure 4 is rotated through some angle β about an axis constructed by drawing a line through the origin connecting the points (090,270) (keeping r_1 and r_2 fixed) Figure 5 is obtained. The angle of rotation is defined as

$$\beta = \min(\theta_1, \theta_2) \qquad (C.10)$$

Point now lies on the rotated polar plane, the position of point b remains fixed, and point c represents the fixed range projection of point b onto the polar plane. Some rather obvious geometric relationships follow immediately;

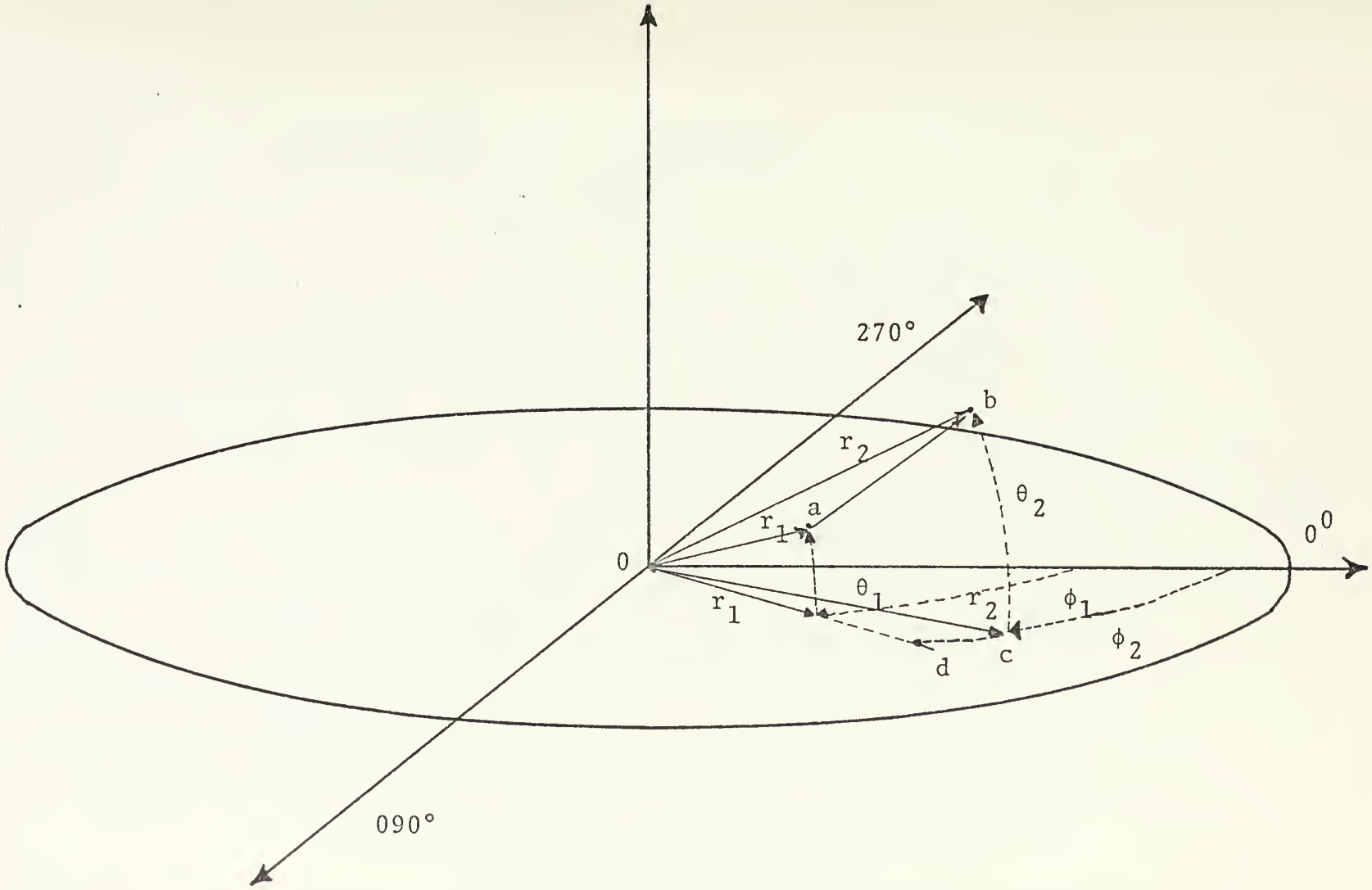


Fig. 4. GEOMETRY OF THE PREDICTED POSITION ERROR PROBLEM IN A SPHERICAL COORDINATE SYSTEM

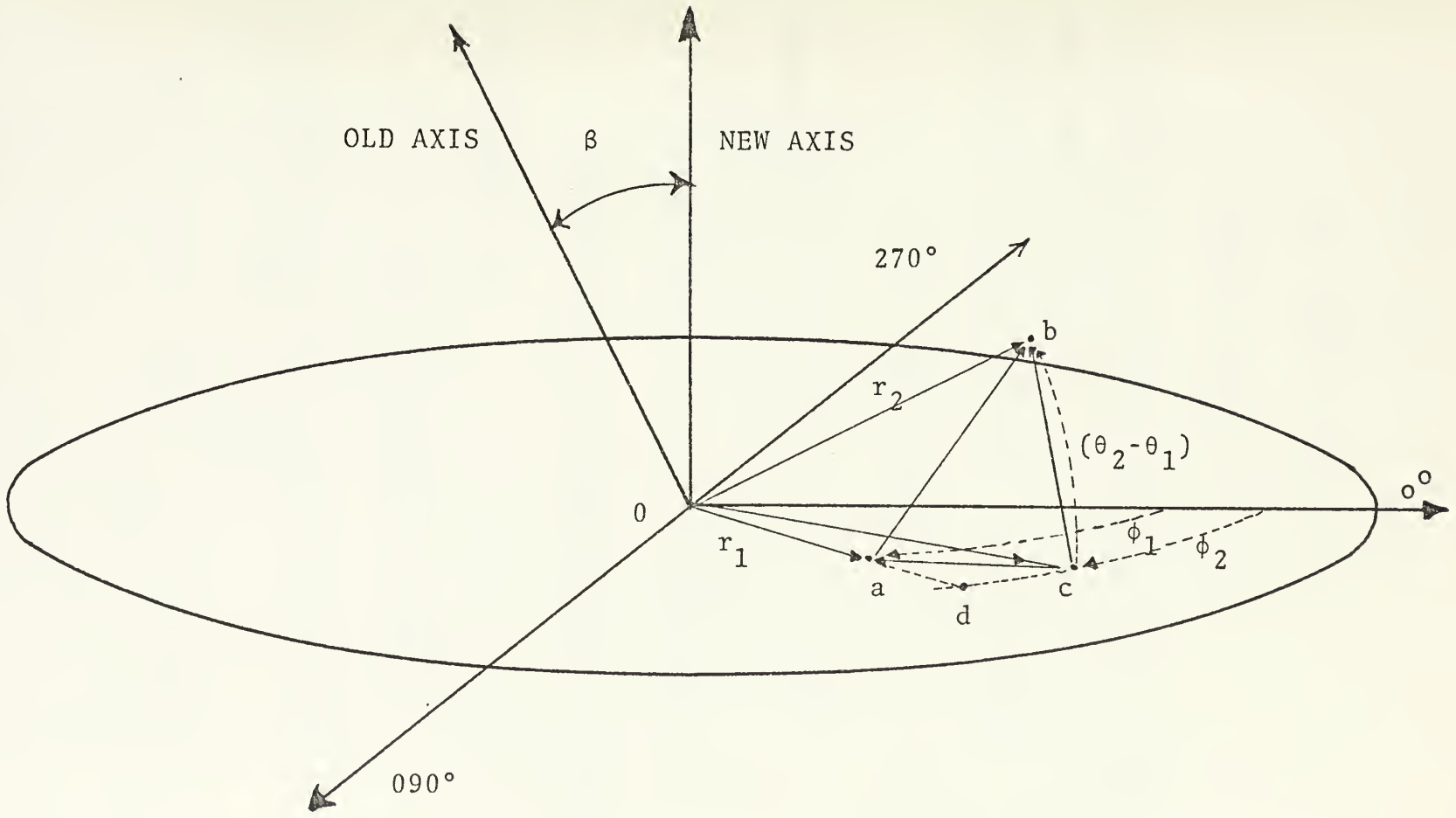


Fig. 5. THE POLAR PLANE AFTER ROTATION

$$\Delta\theta = \underline{/boc} = |\theta_1 - \theta_2|$$

$$\Delta R = |\overrightarrow{ad}| = |r_1 - r_2| \qquad \text{(C.11)}$$

$$|\overrightarrow{od}| = \max(r_1, r_2)$$

The problem then remains of finding |ab|. Figure 6 shows the projection onto the rotated polar plane. Note that triangle cod is an isosceles triangle, then

$$\underline{/cdo} = \frac{180 - \Delta\phi}{2} \qquad \text{(C.12)}$$

Applying the law of cosines $|\overrightarrow{ac}|$ is given by

$$|\overrightarrow{ac}| = \{(|\overrightarrow{ad}|)^2 + [(2.0)(\sin\tfrac{\Delta\phi}{2.0})(|\overrightarrow{od}|)]^2 - (2.0)(|\overrightarrow{ad}|)(2.0)(\sin\tfrac{\Delta\phi}{2.0})(|\overrightarrow{od}|)\cos\underline{/cdo})\}^{1/2} \qquad \text{(C.13)}$$

$|\overrightarrow{bc}|$ is given by

$$|\overrightarrow{bc}| = (2.0)(|\overrightarrow{od}|)(\sin\tfrac{\Delta\theta}{2.0}) \qquad \text{(C.14)}$$

Again applying the law of cosines and noting that $\underline{/bcd} = \underline{/bco}$, $|\overrightarrow{ab}|$ is given by

$$|\overrightarrow{ab}| = \{|\overrightarrow{bc}|^2 + |\overrightarrow{ac}|^2 - (2.0)(|\overrightarrow{bc}|)(|\overrightarrow{ac}|)\cos(\underline{/bcd})\}^{1/2} \qquad \text{(C.15)}$$

Thus, the magnitude of the N-step prediction error has been determined without the need for any coordinate conversion. The validity and accuracy of this means of determining the

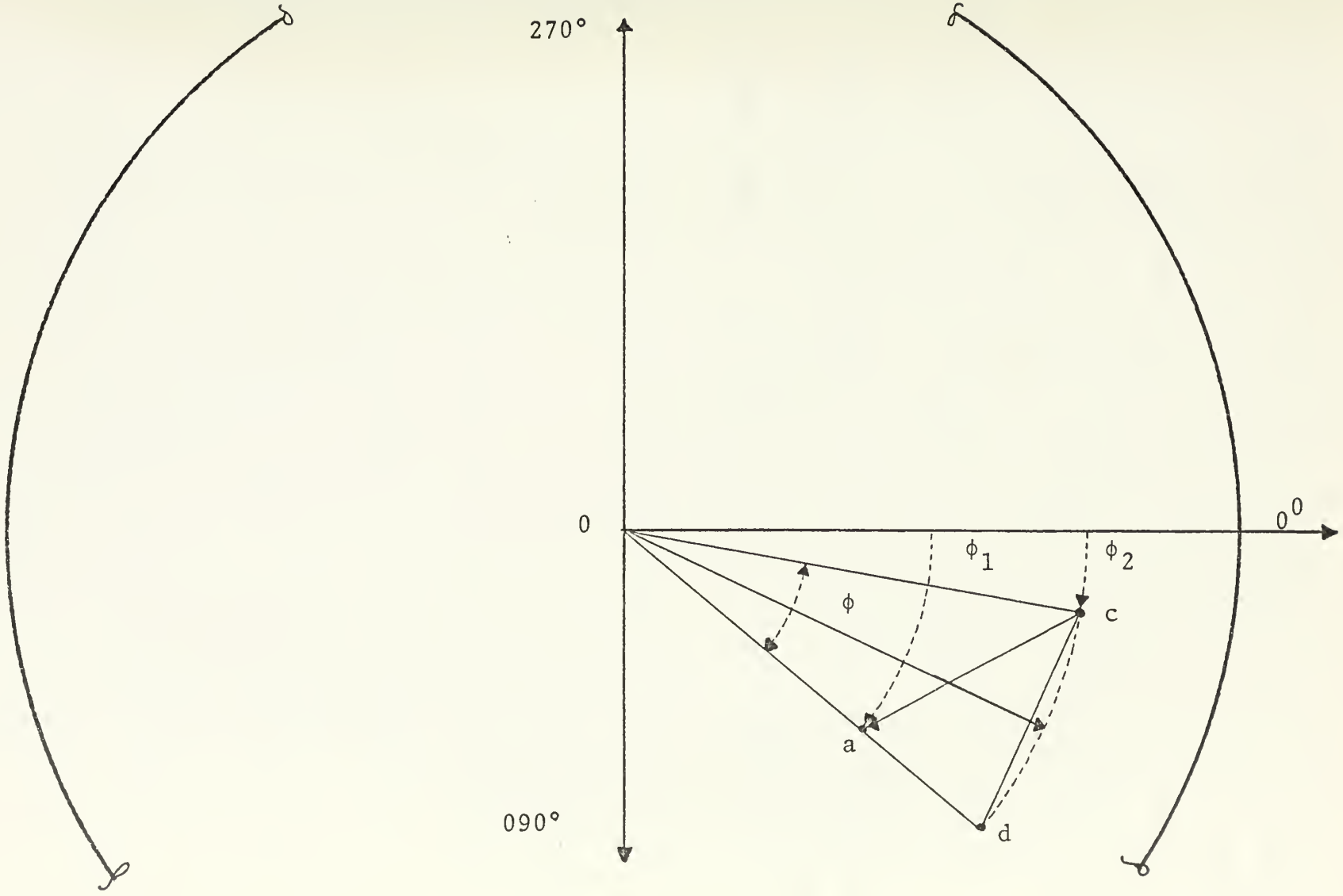


Fig. 6. FIXED RANGE PROJECTION ONTO THE ROTATED POLAR PLANE

error magnitude was verified by conversion of $\underset{\sim}{x}(K+N)$ and and $\hat{\underset{\sim}{x}}(K+N/K)$ to the Cartesian Coordinate system and determining the magnitude of the error vector by standard means. In every case investigated, the results given by the two different methods were in complete agreement. Again note that no attempt is made to determine the direction of the error vector $|\overrightarrow{ab}|$ since the regions of interest specify concentric spheres of constant radii centered at $\underset{\sim}{x}(K+N)$. The directionality of the error vector is eliminated in SUBROUTINE KILL through the use of the absolute values of the differences in bearing and elevation, and by using the maximum value of the actual and estimated ranges.

BIBLIOGRAPHY

1. NAVAL POSTGRADUATE SCHOOL Report NPS-52DE8041A, Adaptive Tracking of Maneuvering Targets, by J. S. Demetry and H. A. Titus, 15 April 1968.

2. NAVAL POSTGRADUATE SCHOOL Report NPS-52KI74101, Evaluation of State Estimators and Predictors for Fire Control Systems, by D. E. Kirk, October 1974.

3. Ketron, M. G., Monte Carlo Evaluation of Digital Filters for Fire Control Systems, M.S. Thesis, NAVAL POSTGRADUATE SCHOOL, June 1974.

4. Parr, J. M., An Estimator for an Anti-Aircraft Gun Fire Control System, M.S. Thesis, NAVAL POSTGRADUATE SCHOOL, December 1974.

5. McKinley, D. H., Modeling Maneuvering Airborne Target with a Coloring Noise Filter, M.S. Thesis, NAVAL POSTGRADUATE SCHOOL, June 1974.

6. Iida, T., Monte Carlo Evaluation of Digital Filters for Fire Control Systems, M.S. Thesis, NAVAL POSTGRADUATE SCHOOL, December 1975.

7. Gelb, A., ed., Applied Optimal Estimation, MIT Press, 1974.

8. Jazwinski, A. H., Stochastic Processes and Filtering Theory, Academic Press, 1970.

9. Sorenson, H. W., "Kalman Filtering Techniques," Advances in Control Systems, C.T. Leonedes, ed., Academic Press, vol. 3, 1966.

10. Singer, R. A., "Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets," IEEE Transactions on Aerospace and Electronic Systems, vol. 6, July 1970.

11. THE ANALYTICAL SCIENCES CORPORATION Report TR-387-1, Adaptive Tracking Filter Design and Evaluation for Gun Fire Control Systems, by C. F. Price and C. M. Brown, 23 January 1974.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 52 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	2
4.	Professor D. E. Kirk, Code 52 Ki Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	3
5.	LT Robert K. Brands 2173 Stanford Avenue St. Paul, Minnesota 55105	2
6.	Mr. H. A. Parish System Integration and Engineering Division Code 4320 Naval Ship Weapon Systems Engineering Station Port Hueneme, California 93043	1
7.	Dr. B. L. Clark System Integration Division Naval Surface Weapons Center Dahlgren, Virginia	1
8.	Mr. C. F. Price The Analytical Sciences Corporation 6 Jacob Way Reading, Massachusetts 01867	1
9.	Professor H. A. Titus, Code 52 Ts Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1

Thesis
B7988
c.1
Brands
An alternative to nonlinear estimation in gun fire control systems.